Computational Linguistics

Models, Resources, Applications

Igor Bolshakov
Alexander Gelbukh



COMPUTACIÓN





CIENCIA DE LA COMPUTACIÓN

COMPUTATIONAL LINGUISTICS
Models, Resources, Applications

COMPUTATIONAL LINGUISTICS
Models, Resources, Applications

Igor A. Bolshakov and Alexander Gelbukh

FIRST EDITION: 2004

ISBN: 970-36-0147- 2

Impreso en México / *Printed in Mexico*

Serie Ciencia de la Computación

The growth of the amount of available written information originated in the Renaissance with the invention of printing press and increased nowadays to unimaginable extent has obliged the man to acquire a new type of literacy related to the new forms of media besides writing. One of such forms is the computer—an object of the modern world that increases the degree of freedom of human action and knowledge, where the fantasy becomes reality, and the new common alphabet penetrates the presence marked by such a phenomenon as computing.

However, even though this phenomenon has become a part of our everyday life, the printed text has not been substituted by the electronic text; on the contrary, they have become into symbiotic elements that constitute fundamental means for accelerating the transition to the advance society and economy restructured towards the science, technology, and promotion and dissemination of knowledge. Only through such spread of knowledge is it possible to create a scientific culture founded on the permanent quest for the truth, informed criticism, and the systematic, rigorous, and intelligent way of human actions.

In this context, the Computer Science Series published by the Center for Computing Research (CIC) of the National Polytechnic Institute in collaboration with the National Autonomous University of Mexico and the Economic Culture Fund editorial house (Fondo de Cultura Económica) presents the works by outstanding Mexican and foreign specialists—outstanding both in their research and educational achievements—in the areas of tutoring systems, system modeling and simulation, numerical analysis, information systems, software engineering, geoprocessing, digital systems, electronics, automatic control, pattern recognition and image processing, natural language processing and artificial intelligence.

In this way, the publishing effort of the CIC—which includes the journal Computación y Sistemas, the Research on Computing Science series, the technical reports, conference proceedings, catalogs of solutions, and this book series—reaffirms its adherence to the high standards of research, teaching, industrial collaboration, guidance, knowledge dissemination, and development of highly skilled human resources.

This series is oriented to specialists in the field of computer science, with the idea to help them to extend and keep up to date their information in this dynamic area of knowledge. It is also intended to be a source of reference in their everyday research and teaching work. In this way one can develop himself or herself basing on the fundamental works of the scientific community—which promotion and dissemination of science is.

We believe that each and every book of this series is a must-have part of the library of any professional in computer science and allied areas who consider learning and keeping one's knowledge up to date essential for personal progress and the progress of our country. Helpful support for this can be found in this book series characterized first and foremost by its originality and excellent quality.

Dr. Juan Luis Díaz De León Santiago
Center For Computing Research
Director

CONTENTS OVERVIEW

DETAILED CONTENTS

PREFACE

WHY DID WE DECIDE to propose a new book on computational linguistics? What are the main objectives, intended readers, the main features, and the relationships of this book to various branches of computer science? In this Preface, we will try to answer these questions.

A NEW BOOK ON COMPUTATIONAL LINGUISTICS

The success of modern software for natural language processing impresses our imagination. Programs for orthography and grammar correction, information retrieval from document databases, and translation from one natural language into another, among others, are sold worldwide in millions of copies nowadays.

However, we have to admit that such programs still lack real intelligence. The ambitious goal of creating software for deep language understanding and production, which would provide tools powerful enough for fully adequate automatic translation and man-machine communication in unrestricted natural language, has not yet been achieved, though attempts to solve this problem already have a history of nearly 50 years.

This suggests that in order to solve the problem, developers of new software will need to use the methods and results of a fundamental science, in this case linguistics, rather than the tactics of *ad hoc* solutions. Neither increasing the speed of computers, nor refinement of programming tools, nor further development of numerous toy systems for language "understanding" in tiny domains, will suffice to solve one of the most challenging problems of modern science—automatic text understanding.

We believe that this problem, yet unsolved in the past century, will be solved in the beginning of this century by those who are sit-

ting now on student benches. This book on computational linguistics models and their applications is targeted at these students, namely, at those students of the Latin American universities studying computer science and technology who are interested in the development of natural language processing software.

Thus, we expect the students to have already some background in computer science, though no special training in the humanities and in linguistics in particular.

On the modern book market, there are many texts on Natural Language Processing (NLP), e.g. [1, 2, 7, 9]. They are quite appropriate as the further step in education for the students in computer science interested in the selected field. However, for the novices in linguistics (and the students in computer science are among them) the available books still leave some space for additional manuals because of the following shortages:

- Many of them are English-oriented. Meanwhile, English, in spite of all its vocabulary similarities with Spanish, is a language with quite a different grammatical structure. Unlike Spanish, English has a very strict word order and its morphology is very simple, so that the direct transfer of the methods of morphologic and syntactic analysis from English to Spanish is dubious.

- Only few of these manuals have as their objective to give a united and comparative exposition of various coexisting theories of text processing. Moreover, even those few ones are not very successful since the methodologies to be observed and compared are too diverse in their approaches. Sometimes they even contradict each other in their definitions and notations.

- The majority of these manuals are oriented only to the formalisms of syntax, so that some of them seemingly reduce computational linguistics to a science about English syntax. Nevertheless, linguistics in general investigates various linguistic levels, namely, phonology, morphology, syntax, and semantics. For each of these levels, the amount of purely linguistic knowledge rele-

vant for computational linguistics seems now much greater than that represented in well-known manuals on the subject.

Reckoning with all complications and controversies of the quickly developing discipline, the main methodological features of this book on computational linguistics are the following:

- Nearly all included examples are taken from Spanish. The other languages are considered mainly for comparison or in the cases when the features to be illustrated are not characteristic to Spanish.
- A wide variety of the facts from the fundamental theory—general linguistics—that can be relevant for the processing of natural languages, right now or in the future, are touched upon in one way or another, rather than only the popular elements of English-centered manuals.
- Our educational line combines various approaches coexisting in computational linguistics and coordinates them wherever possible.
- Our exposition of the matter is rather slow and measured, in order to be understandable for the readers who do not have any background in linguistics.

In fact, we feel inappropriate to simply gather disjoint approaches under a single cover. We also have rejected the idea to make our manual a reference book, and we do not have the intention to give always well-weighted reviews and numerous references through our texts. Instead, we consider the coherence, consistency, and self-containment of exposition to be much more important.

The two approaches that most influenced the contents of this book are the following:

- The Meaning ⇔ Text Theory (MTT), developed by Igor Mel'čuk, Alexander Žolkovsky, and Yuri Apresian since the mid-sixties, facilitates describing the elements, levels, and structures of natural languages. This theory is quite appropriate for any language,

but especially suits for languages with free word order, including Spanish. Additionally, the MTT gives an opportunity to validate and extend the traditional terminology and methodology of linguistics.

- The *Head-driven Phrase Structure Grammar* (HPSG), developed by Carl Pollard and Ivan Sag in the last decade, is probably the most advanced practical formalism in natural language description and processing within the modern tradition of generative grammars originated by Noam Chomsky. Like the MTT, HPSG takes all known facts for description of natural languages and tries to involve new ones. As most of the existing formalisms, this theory was mainly tested on English. In recent years, however, HPSG has acquired numerous followers among researchers of various languages, including Spanish. Since the main part of the research in NLP has been fulfilled till now in the Chomskian paradigm, it is very important for a specialist in computational linguistics to have a deeper knowledge of the generative grammar approach.

The choice of our material is based on our practical experience and on our observations on the sources of the problems which we ourselves and our colleagues encountered while starting our careers in computational linguistics and which still trouble many programmers working in this field due to the lack of fundamental linguistic knowledge.

After coping with this book, our reader would be more confident to begin studying such branches of computational linguistics as

- Mathematical Tools and Structures of Computational Linguistics,
- Phonology,
- Morphology,
- Syntax of both surface and deep levels, and
- Semantics.

The contents of the book are based on the course on computational linguistics that has been delivered by the authors since 1997

at the Center for Computing Research, National Polytechnic Institute, Mexico City. This course was focused on the basic set of ideas and facts from the fundamental science necessary for the creation of intelligent language processing tools, without going deeply into the details of specific algorithms or toy systems. The practical study of algorithms, architectures, and maintenance of real-world applied linguistic systems may be the topics of other courses.

Since most of the literature on this matter is published in English regardless of the country where the research was performed, it will be useful for the students to read an introduction to the field in English. However, Spanish terminological equivalents are also given in the Appendix (see page 173).

The book is also supplied with 54 review questions, 58 test questions recommended for the exam, with 4 variants of answer for each one, 30 illustrations, 58 bibliographic references, and 37 references to the most relevant Internet sites.

The authors can be contacted at the following e-mail addresses: igor@cic.ipn.mx, gelbukh@gelbukh.com (gelbukh@cic.ipn.mx); see also www.Gelbukh.com (www.cic.ipn.mx/~gelbukh). The webpage for this book is www.Gelbukh.com/clbook.

OBJECTIVES AND INTENDED READERS OF THE BOOK

The main objectives of this book are to provide the students with few fundamentals of general linguistics, to describe the modern models of how natural languages function, and to explain how to compile the data—linguistic tables and machine dictionaries—necessary for the natural language processing systems, out of informally described facts of a natural language. Therefore, we want to teach the reader how to prepare all the necessary tools for the development of programs and systems oriented to automatic natural language processing. In order to repeat, we assume that our readers are mainly students in computer sciences, i.e., in software development, database management, information retrieval, artificial intelligence or computer science in general.

Throughout this book, special emphasis is made on applications to the Spanish language. However, this course is not a mere manual of Spanish. A broader basis for understanding the main principles is to be elucidated through some examples from English, French, Portuguese, and Russian. Many literature sources provide the reader with interesting examples for these languages. In our books, we provide analogous examples for Spanish wherever possible.

Significant difficulties were connected with the fact that Latin American students of technical institutes have almost no knowledge in linguistics beyond some basics of Spanish grammar they learned in their primary schooling, at least seven years earlier. Therefore, we have tried to make these books understandable for students without any background in even rather elementary grammar.

Neither it should be forgotten that the native language is studied in the school prescriptively, i.e., how it is preferable or not recommendable to speak and write, rather than descriptively, i.e., how the language is really structured and used.

However, only complete scientific description can separate correct or admissible language constructions from those not correct and not belonging to the language under investigation. Meantime, without a complete and correct description, computer makes errors quite unusual and illogical from a human point of view, so that the problem of text processing cannot be successfully solved.

COORDINATION WITH COMPUTER SCIENCE

The emphasis on theoretical issues of language in this book should not be understood as a lack of coordination between computational linguistics and computer science in general. Computer science and practical programming is a powerful tool in all fields of information processing. Basic knowledge of computer science and programming is expected from the reader.

The objective of the book is to help the students in developing applied software systems and in choosing the proper models and data structures for these systems. We only reject the idea that the

computer science's tools of recent decades are sufficient for computational linguistics in theoretical aspects. Neither proper structuring of linguistic programs, nor object-oriented technology, nor specialized languages of artificial intelligence like Lisp or Prolog solve by themselves the problems of computational linguistics. All these techniques are just tools.

As it is argued in this book, the ultimate task of many applied linguistic systems is the transformation of an unprepared, unformatted natural language text into some kind of representation of its meaning, and, vice versa, the transformation of the representation of meaning to a text. It is the main task of any applied system.

However, the significant part of the effort in the practical developing of an NPL system is *not* connected directly with creating the software for this ultimate task. Instead, more numerous, tedious, and inevitable programming tasks are connected with *extraction of data* for the grammar tables and machine dictionaries from various texts or human-oriented dictionaries. Such texts can be originally completely unformatted, partially formatted, or formalized for some other purposes. For example, if we have a typical human-oriented dictionary, in the form of a text file or database, our task can be to parse each dictionary entry and to extract all of the data necessary for the ultimate task formulated above.

This book contains the material to learn how to routinely solve such tasks. Thus, again, we consider programming to be the everyday practical tool of the reader and the ultimate goal of our studies.

COORDINATION WITH ARTIFICIAL INTELLIGENCE

The links between computational linguistics and artificial intelligence (AI) are rather complicated. Those AI systems that contain subsystems for natural language processing rely directly on the ideas and methods of computational linguistics. At the same time, some methods usually considered belonging only to AI, such as, for example, algorithms of search for decision in trees, matrices, and

other complex structures, sometimes with backtracking, are applicable also to linguistic software systems.

Because of this, many specialists in AI consider computational linguistics a part of AI [2, 40, 54]. Though such an expansion hampers nothing, it is not well grounded, in our opinion, since computational linguistics has its own theoretical background, scientific neighbors, methods of knowledge representation, and decision search. The sample systems of natural language processing, which wander from one manual on AI to another, seem rather toy and obsolete.

Though the two fields of research are different, those familiar with both fields can be more productive. Indeed, they do not need to invent things already well known in the adjacent field, just taking from the neighborhood what they prefer for their own purposes. Thus, we encourage our readers to deeply familiarize themselves with the area of AI. We also believe that our books could be useful for students specializing in AI.

SELECTION OF TOPICS

Since the MTT described below in detail improves and enriches rather than rejects the previous tradition in general linguistics, we will mainly follow the MTT in description and explanation of facts and features of natural languages, giving specific emphasis on Spanish.

To avoid a scientific jumble, we usually do not mark what issues are characteristic to the MTT, but are absent or are described in a different way in other theories. We give in parallel the point of view of the HPSG-like formalisms on the same issue only in the places where it is necessary for the reader to be able to understand the corresponding books and articles on computational linguistics written in the tradition originated by Chomsky.

We should assert here that the MTT is only a tool for language description. It does not bind researchers to specific algorithms or to specific formats of representation of linguistic data. We can find the

same claims about a purely linguistic orientation in the recent books on the HPSG as well, though the latter approach seems more computer-oriented. In prospect, we hope, those two approaches will complement and enrich each other.

WEB RESOURCES FOR THIS BOOK

The webpage of this book is www.Gelbukh.com/clbook. You can find there additional materials on the topics of this book, links to relevant Internet resources, and errata. Many publications by the authors of this book can be found at www.Gelbukh.com.

ACKNOWLEDGMENTS

We are profoundly grateful to Prof. Igor Mel'čuk for placing in our disposal his recent works and for reading through a nearly ready version of the book, with a lot of bitter criticism. We are grateful also to Prof. Yuri Apresian and his research team in Moscow, especially to Prof. Leonid Tsinman and Prof. Igor Boguslavsky, for providing us with the materials on the applications of the Meaning ⇔ Text Theory and for some discussions. Dr. Patrick Cassidy of MICRA, Inc., has read parts of the manuscript and provided us with helpful advice.

We express our most cordial thanks to our doctoral student Sofía N. Galicia Haro, who performed a lot of administrative chores and paperwork, freeing our time for the work on the book. She also was our main advisor in everything concerning Spanish language.

Finally, we thank our families for their great patience and constant support.

I. INTRODUCTION

IS IT NECESSARY to automatically process natural language texts, if we can just read them? Moreover, is it difficult anyway, when every child easily learns how to read in primary school? Well, if it *is* difficult, is it then possible at all? What do we need to know in order to develop a computer program that would do it? What parts of linguistics are most important for this task?

In this introductory chapter, we will answer these questions, and the main answers will be: yes, it *is* necessary; yes, it is *very* difficult; and yes, it *is* possible.

THE ROLE OF NATURAL LANGUAGE PROCESSING

We live in the age of information. It pours upon us from the pages of newspapers and magazines, radio loudspeakers, TV and computer screens. The main part of this information has the form of natural language texts. Even in the area of computers, a larger part of the information they manipulate nowadays has the form of a text. It looks as if a personal computer has mainly turned into a tool to create, proofread, store, manage, and search for text documents.

Our ancestors invented natural language many thousands of years ago for the needs of a developing human society. Modern natural languages are developing according to their own laws, in each epoch being an adequate tool for human communication, for expressing human feelings, thoughts, and actions. The structure and use of a natural language is based on the assumption that the participants of the conversation share a very similar experience and knowledge, as well as a manner of feeling, reasoning, and acting. The great challenge of the problem of intelligent automatic text processing is to use unrestricted natural language to exchange information with a creature of a totally different nature: the computer.

For the last two centuries, humanity has successfully coped with the automation of many tasks using mechanical and electrical devices, and these devices faithfully serve people in their everyday life. In the second half of the twentieth century, human attention has turned to the automation of natural language processing. People now want assistance not only in mechanical, but also in intellectual efforts. They would like the machine to read an unprepared text, to test it for correctness, to execute the instructions contained in the text, or even to comprehend it well enough to produce a reasonable response based on its meaning. Human beings want to keep for themselves only the final decisions.

The necessity for intelligent automatic text processing arises mainly from the following two circumstances, both being connected with the quantity of the texts produced and used nowadays in the world:

- Millions and millions of persons dealing with texts throughout the world do not have enough knowledge and education, or just time and a wish, to meet the modern standards of document processing. For example, a secretary in an office cannot take into consideration each time the hundreds of various rules necessary to write down a good business letter to another company, especially when he or she is not writing in his or her native language. It is just cheaper to teach the machine once to do this work, rather than repeatedly teach every new generation of computer users to do it by themselves.

- In many cases, to make a well-informed decision or to find information, one needs to read, understand, and take into consideration a quantity of texts thousands times larger than one person is physically able to read in a lifetime. For example, to find information in the Internet on, let us say, the expected demand for a specific product in the next month, a lot of secretaries would have to read texts for a hundred years without eating and sleeping, looking through all the documents where this information

might appear. In such cases, using a computer is the only possible way to accomplish the task.

Thus, the processing of natural language has become one of the main problems in information exchange. The rapid development of computers in the last two decades has made possible the implementation of many ideas to solve the problems that one could not even imagine being solved automatically, say, 45 years ago, when the first computers appeared.

Intelligent natural language processing is based on the science called computational linguistics. Computational linguistics is closely connected with applied linguistics and linguistics in general. Therefore, we shall first outline shortly linguistics as a science belonging to the humanities.

LINGUISTICS AND ITS STRUCTURE

Linguistics is a science about natural languages. To be more precise, it covers a whole set of different related sciences (see Figure I.1).

General linguistics is a nucleus [18, 36]. It studies the general structure of various natural languages and discovers the universal laws of functioning of natural languages. Many concepts from general linguistics prove to be necessary for any researcher who deals with natural languages. General linguistics is a fundamental science that was developed by many researchers during the last two centuries, and it is largely based on the methods and results of grammarians of older times, beginning from the classical antiquity.

As far as general linguistics is concerned, its most important parts are the following:

- *Phonology* deals with sounds composing speech, with all their similarities and differences permitting to form and distinguish words.

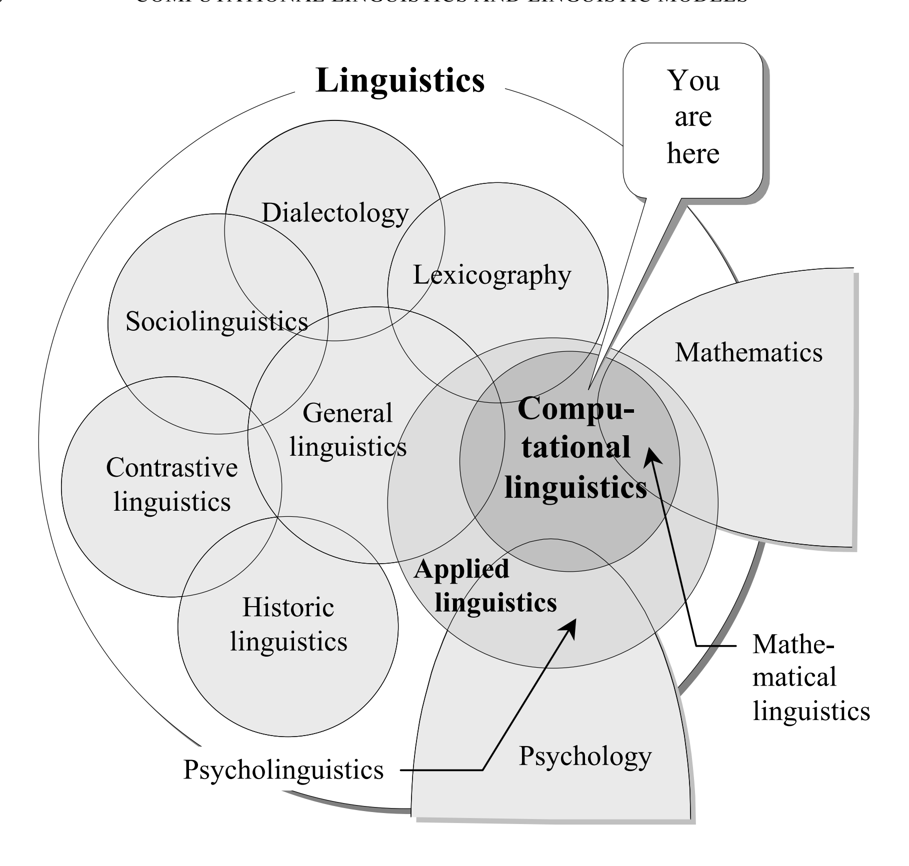


FIGURE I.1. *Structure of linguistic science.*

- *Morphology* deals with inner structure of individual words and the laws concerning the formation of new words from pieces—morphs.
- *Syntax* considers structures of sentences and the ways individual words are connected within them.
- *Semantics* and *pragmatics* are closely related. Semantics deals with the meaning of individual words and entire texts, and pragmatics studies the motivations of people to produce specific sentences or texts in a specific situation.

There are many other, more specialized, components of linguistics as a whole (see Figure I.1).

Historical, or comparative, linguistics studies history of languages by their mutual comparison, i.e., investigating the history of their similarities and differences. The second name is explained by the fact that comparison is the main method in this branch of linguistics. Comparative linguistics is even older than general linguistics, taking its origin from the eighteenth century.

Many useful notions of general linguistics were adopted directly from comparative linguistics.

From the times of Ferdinand de Saussure, the history of language has been called *diachrony* of language, as opposed to the *synchrony* of language dealing with phenomena of modern natural languages only. Therefore, diachrony describes changes of a language along the time axis.

Historical linguistics discovered, for example, that all Romance languages (Spanish, Italian, French, Portuguese, Romanian, and several others) are descendants of Latin language. All languages of the Germanic family (German, Dutch, English, Swedish, and several others) have their origins in a common language that was spoken when German tribes did not yet have any written history. A similar history was discovered for another large European family of languages, namely, for Slavonic languages (Russian, Polish, Czech, Croatian, Bulgarian, among others).

Comparative study reveals many common words and constructions within each of the mentioned families—Romance, Germanic, and Slavonic—taken separately.

At the same time, it has noticed a number of similar words among these families. This finding has led to the conclusion that the mentioned families form a broader community of languages, which was called Indo-European languages. Several thousand years ago, the ancestors of the people now speaking Romance, Germanic, and Slavonic languages in Europe probably formed a common tribe or related tribes.

At the same time, historic studies permits to explain why English has so many words in common with the Romance family, or why Romanian language has so many Slavonic words (these are referred to as *loan words*).

Comparative linguistics allows us to predict the elements of one language based on our knowledge of another related language. For example, it is easy to guess the unknown word in the following table of analogy:

Spanish	English
constitución	*constitution*
revolución	*revolution*
investigación	?

Based on more complicated phonologic laws, it is possible even to predict the pronunciation of the French word for the Spanish *agua* (namely [o], *eau* in the written form), though at the first glance these two words are quite different (actually, both were derived from the Latin word *aqua*).

As to computational linguistics, it can appeal to diachrony, but usually only for motivation of purely synchronic models. History sometimes gives good suggestions for description of the current state of language, helping the researcher to understand its structure.

Contrastive linguistics, or *linguistic typology,* classifies a variety of languages according to the similarity of their features, notwithstanding the origin of languages. The following are examples of classification of languages not connected with their origin.

Some languages use articles (like *a* and *the* in English) as an auxiliary *part of speech* to express definite/indefinite use of nouns. (Part of speech is defined as a large group of words having some identical morphologic and syntactic properties.) Romance and Germanic languages use articles, as well as Bulgarian within the Slavonic family. Meantime, many other languages do not have articles

(nearly all Slavonic family and Lithuanian, among others). The availability of articles influences some other features of languages.

Some languages have the so-called grammatical cases for several parts of speech (nearly all Slavonic languages, German, etc.), whereas many others do not have them (Romance languages, English—from the Germanic family, Bulgarian—from the Slavonic family, and so on).

Latin had nominative (direct) case and five oblique cases: genitive, dative, accusative, ablative, and vocative. Russian has also six cases, and some of them are rather similar in their functions to those of Latin. Inflected parts of speech, i.e., nouns, adjectives, participles, and pronouns, have different word endings for each case.

In English, there is only one oblique case, and it is applicable only to some personal pronouns: *me, us, him, her, them.*

In Spanish, two oblique cases can be observed for personal pronouns, i.e., dative and accusative: *le, les, me, te, nos, las,* etc. Grammatical cases give additional mean for exhibiting syntactic dependencies between words in a sentence. Thus, the inflectional languages have common syntactic features.

In a vast family of languages, the main type of sentences contains a *syntactic subject* (usually it is the agent of an action), a *syntactic predicate* (usually it denotes the very action), and a *syntactic object* (usually it is the target or patient of the action). The subject is in a standard form (i.e., in direct, or nominative, case), whereas the object is usually in an oblique case or enters in a prepositional group. This is referred to as non-ergative construction.

Meantime, a multiplicity of languages related to various other families, not being cognate to each other, are classified as ergative languages. In a sentence of an ergative language, the agent of the action is in a special oblique (called ergative) case, whereas the object is in a standard form. In some approximation, a construction similar to an ergative one can be found in the Spanish sentence *Me simpatizan los vecinos*, where the real agent (feeler) *yo* 'I' is used in oblique case *me*, whereas the object of feeling, *vecinos*, stays in the standard form. All ergative languages are considered typologically

similar to each other, though they might not have any common word. The similarity of syntactical structures unites them in a common typological group.

Sociolinguistics describes variations of a language along the social scale. It is well known that various social strata often use different sublanguages within the same common language, wherever the same person uses different sublanguages in different situations. It suffices to compare the words and their combinations you use in your own formal documents and in conversations with your friends.

Dialectology compares and describes various dialects, or sublanguages, of a common language, which are used in different areas of the territory where the same language is officially used. It can be said that dialectology describes variations of a language throughout the space axis (while diachrony goes along the time axis). For example, in different Spanish-speaking countries, many words, word combinations, or even grammatical forms are used differently, not to mention significant differences in pronunciation. Gabriel García Márquez, the world-famous Colombian writer, when describing his activity as a professor at the International Workshop of cinematographers in Cuba, said that it was rather difficult to use only the words common to the entire Spanish-speaking world, to be equally understandable to all his pupils from various countries of Latin America. A study of Mexican Spanish, among other variants of Spanish language is a good example of a task in the area of dialectology.

Lexicography studies the lexicon, or the set of all words, of a specific language, with their meanings, grammatical features, pronunciation, etc., as well as the methods of compilation of various dictionaries based on this knowledge. The results of lexicography are very important for many tasks in computational linguistics, since any text consists of words. Any automatic processing of a text starts with retrieving the information on each word from a computer dictionary compiled beforehand.

Psycholinguistics studies the language behavior of human beings by the means of a series of experiments of a psychological type.

Among areas of its special interest, psycholinguists studies teaching language to children, links between the language ability in general and the art of speech, as well as other human psychological features connected with natural language and expressed through it. In many theories of natural language processing, data of psycholinguistics are used to justify the introduction of the suggested methods, algorithms, or structures by claiming that humans process language "just in this way."

Mathematical linguistics. There are two different views on mathematical linguistics. In the narrower view, the term *mathematical linguistics* is used for the theory of formal grammars of a specific type referred to as *generative grammars.* This is one of the first purely mathematical theories devoted to natural language. Alternatively, in the broader view, mathematical linguistics is the intersection between linguistics and mathematics, i.e., the part of mathematics that takes linguistic phenomena and the relationships between them as the objects of its possible applications and interpretations.

Since the theory of generative grammars is nowadays not unique among linguistic applications of mathematics, we will follow the second, broader view on mathematical linguistics.

One of the branches of mathematical linguistics is *quantitative linguistic.* It studies language by means of determining the frequencies of various words, word combinations, and constructions in texts. Currently, quantitative linguistics mainly means *statistical linguistics*. It provides the methods of making decisions in text processing on the base of previously gathered statistics.

One type of such decisions is resolution of ambiguity in text fragments to be analyzed. Another application of statistical methods is in the deciphering of texts in forgotten languages or unknown writing systems. As an example, deciphering of Mayan glyphs was fulfilled in the 1950's by Yuri Knorozov [39] taking into account statistics of different glyphs (see Figure I.2).

Applied linguistics develops the methods of using the ideas and notions of general linguistics in broad human practice. Until the



FIGURE I.2. *The ancient Mayan writing system was deciphered with statistical methods.*

middle of the twentieth century, applications of linguistics were limited to developing and improving grammars and dictionaries in a printed form oriented to their broader use by non-specialists, as well as to the rational methods of teaching natural languages, their orthography and stylistics. This was the only purely practical product of linguistics.

In the latter half of the twentieth century, a new branch of applied linguistics arose, namely the *computational*, or *engineering*, linguis-

tics. Actually, this is the main topic of this book, and it is discussed in some detail in the next section.

WHAT WE MEAN BY COMPUTATIONAL LINGUISTICS

Computational linguistics might be considered as a synonym of automatic processing of natural language, since the main task of computational linguistics is just the construction of computer programs to process words and texts in natural language.

The processing of natural language should be considered here in a very broad sense that will be discussed later.

Actually, this course is slightly "more linguistic than computational," for the following reasons:

- We are mainly interested in the formal description of language relevant to automatic language processing, rather than in purely algorithmic issues. The algorithms, the corresponding programs, and the programming technologies can vary, while the basic linguistic principles and methods of their description are much more stable.
- In addition to some purely computational issues, we also touch upon the issues related to computer science only in an indirect manner. A broader set of notions and models of general linguistics and mathematical linguistics are described below.

For the purposes of this course, it is also useful to draw a line between the issues in text processing we consider linguistic—and thus will discuss below—and the ones we will not. In our opinion, for a computer system or its part to be considered linguistic, it should use some data or procedures that are:

- *language-dependent*, i.e., change from one natural language to another,
- *large*, i.e., require a significant amount of work for compilation.

Thus, not every program dealing with natural language texts is related to linguistics. Though such word processors as Windows' Notebook do deal with the processing of texts in natural language, we do not consider them linguistic software, since they are not sufficiently language-dependent: they can be used equally for processing of Spanish, English, or Russian texts, after some alphabetic adjustments.

Let us put another example: some word processors can hyphenate words according to the information about the *vowels* and *consonants* in a specific alphabet and about *syllable* formation in a specific language. Thus, they *are* language-dependent. However, they do not rely on large enough linguistic resources. Therefore, simple hyphenation programs only border upon the software that can be considered linguistic proper. As to spell checkers that use a large word list and complicated morphologic tables, they are just linguistic programs.

WORD, WHAT IS IT?

As it could be noticed, the term *word* was used in the previous sections very loosely. Its meaning seems obvious: any language operates with words and any text or utterance consists of them. This notion seems so simple that, at the first glance, it does not require any strict definition or further explanation: one can think that a word is just a substring of the text as a letter string, from the first delimiter (usually, a space) to the next one (usually, a space or a punctuation mark). Nevertheless, the situation is not so simple.

Let us consider the Spanish sentence *Yo devuelvo los libros el próximo mes, pero tú me devuelves el libro ahora*. How many words does it contain? One can say 14 and will be right, since there are just 14 letter substrings from one delimiter to another in this sentence. One can also notice that the article *el* is repeated twice, so that the number of different words (substrings) is 13. For these observations, no linguistic knowledge is necessary.

However, one can also notice that *devuelvo* and *devuelves* are forms of the same verb *devolver*, and *libros* and *libro* are forms of the same noun *libro*, so that the number of different words is only 11. Indeed, these pairs of wordforms denote the same action or thing. If one additionally notices that the article *los* is essentially equivalent to the article *el* whereas the difference in grammatical number is ignored, then there are only 10 different words in this sentence. In all these cases, the "equivalent" strings are to some degree similar in their appearance, i.e., they have some letters in common.

At last, one can consider *me* the same as *yo*, but given in oblique grammatical case, even though there are no letters in common in these substrings. For such an approach, the total number of different words is nine.

We can conclude from the example that the term *word* is too ambiguous to be used in a science with the objective to give a precise description of a natural language. To introduce a more consistent terminology, let us call an individual substring used in a specific place of a text (without taking into account its possible direct repetitions or similarities to other substrings) a *word occurrence*. Now we can say that the sentence above consisted of 14 word occurrences.

Some of the substrings (usually similar in the appearance) have the same core meaning. We intuitively consider them as different forms of some common entity. A set of such forms is called *lexeme*. For example, in Spanish {*libro, libros*}, {*alto, alta, altos, altas*}, and {*devolver, devuelvo, devuelves, devuelve, devolvemos*...} are lexemes. Indeed, in each set there is a commonality between the strings in the letters they consist of (the commonality being expressed as patterns *libro-*, *alt-,* and *dev...lv-*), and their meanings are equivalent (namely, 'book', 'high', and 'to bring back', correspondingly). Each entry of such a set—a letter string without regard to its position in the text—is called *wordform*. Each word occurrence represents a wordform, while wordforms (but not word occurrences) can repeat in the text. Now we can say that the sentence in the example above contains 14 word occurrences, 13 different wordforms,

or nine different lexemes. The considerations that gave other figures in the example above are linguistically inconsistent.

A lexeme is identified by a name. Usually, one of its wordforms, i.e., a specific member of the wordform set, is selected for this purpose. In the previous examples, *LIBRO, ALTO,* and *DEVOLVER* are taken as names of the corresponding lexemes. Just these names are used as titles of the corresponding entries in dictionaries mentioned above. The dictionaries cover available information about lexemes of a given language, sometimes including morphologic information, i.e., the information on how wordforms of these lexemes are constructed. Various dictionaries compiled for the needs of lexicography, dialectology, and sociolinguistics have just lexemes as their entries rather than wordforms.

Therefore, the term *word*, as well as its counterparts in other languages, such as Spanish *palabra*, is too ambiguous to be used in a linguistic book. Instead, we should generally use the terms *word occurrence* for a specific string in a specific place in the text, *wordform* for a string regardless to its specific place in any text, and *lexeme* for a theoretical construction uniting several wordforms corresponding to a common meaning in the manner discussed above.

However, sometimes we will retain the habitual word *word* when it is obvious which of these more specific terms is actually meant.

THE IMPORTANT ROLE OF THE FUNDAMENTAL SCIENCE

In the past few decades, many attempts to build language processing or language understanding systems have been undertaken by people without sufficient knowledge in theoretical linguistics. They hoped that they would succeed thanks to clever mathematical algorithms, fine programming in Assembler language, or just the speed of their computers. To our knowledge, all such attempts have failed. Even now it is still worthwhile to explain the necessity to have fundamental knowledge for those who would develop such systems, and thus to clarify why we decided to start a course in computational linguistics from notions of general linguistics.

General linguistics is a fundamental science belonging to the humanities. An analogy with another fundamental science—physics—is appropriate here.

A specialist with deep knowledge of physics would easily understand the structure of any new electrical device.

Moreover, since the fundamentals of physics are changing very slowly, such a specialist would be able to understand those new or revolutionary engineering principles that did not even exist at the time when he or she studied in a university. Indeed, the underlying fundamentals of physics have remained the same.

On the contrary, somebody with narrow specialization only in laser devices might be perplexed with any new principle that does not relate to his or her deep but narrow knowledge.

The same situation can be observed with linguistics. Even experienced programmers who have taken solid courses on software systems for natural language processing might become helpless in cases where a deeper penetration into linguistic notions and laws is needed.

What is more, they will hardly be able to compile a formal grammar, grammar tables, or a computer dictionary of a natural language, whereas the program heavily relies on such tables and dictionaries as on its integral parts.

They will not even be able to understand the suggestions of professional linguists (who regrettably in many cases prove to know little about programming) and will not be able to explain to them the meaning of the data structures the program needs to use.

On the contrary, a good background in linguistics will allow the new specialists in computational linguistics to work productively in an interdisciplinary team, or just to compile all the necessary tables and dictionaries on their own.

It might even guide them to some quite new ideas and approaches. These specialists will better understand the literature on the subject, which is very important in our rapidly developing world.

CURRENT STATE OF APPLIED RESEARCH ON SPANISH

In our books, the stress on Spanish language is made intentionally and purposefully. For historical reasons, the majority of the literature on natural languages processing is not only written in English, but also takes English as the target language for these studies. In our opinion, this is counter-productive and thus it has become one of the causes of a lag in applied research on natural language processing in many countries, compared to the United States. The Spanish-speaking world is not an exception to this rule.

The number of Spanish-speaking people in the world has exceeded now 400 million, and Spanish is one of the official languages of the United Nations. As to the human-oriented way of teaching, Spanish is well described, and the Royal Academy of Madrid constantly supports orthographic [33] and grammatical research [30] and standardization. There are also several good academic-type dictionaries of Spanish, one the best of which being [28].

However, the lexicographic research reflected in these dictionaries is too human-oriented. Along with some historical information, these dictionaries provide semantic explanations, but without a formal description of the main linguistic properties of lexemes, even in morphologic and syntactic aspects.

Formal description and algorithmization of a language is the objective of research teams in computational linguistics. Several teams in this field oriented to Spanish work now in Barcelona and Madrid, Spain. However, even this is rather little for a country of the European Community, where unilingual and multilingual efforts are well supported by the governmental and international agencies. Some research on Spanish is conducted in the United States, for example, at New Mexico State University.

In Mexico—the world's largest Spanish-speaking country—the activity in computational linguistics has been rather low in the past decades. Now, the team headed by Prof. L.A. Pineda Cortés at National Autonomous University of Mexico is working on a very difficult task of creation of a program that will be able to perform a dia-

logue in Spanish with a human. A very useful dictionary of modern Mexican Spanish, developed by the team headed by Prof. L.F. Lara Ramos [26] (see also [47]), is oriented to human users, giving semantic interpretations and suggestions on good usage of words.

Some additional information on Spanish-oriented groups can be found in the Appendix on the page 173.

As to the books by Helena Beristáin [11], Irene Gartz [15], and J.L. Fuentes [14] on Spanish grammar, they are just well structured[1] manuals of language oriented to native speakers, and thus cannot be used directly as a source of grammatical information for a computer program.

One of the most powerful corporations in the world, Microsoft, has announced the development of a natural language processing system for Spanish based on the idea of multistage processing. As usually with commercial developments, the details of the project are still rather scarce. We can only guess that a rather slow progress of the grammar checker of Word text processor for Windows is related somehow with these developments.

Thus, one who needs to compile all facts of Spanish relevant for its automatic processing faces with a small set of rather old monographs and manuals oriented to human learners, mainly written and first published in Spain and then sometimes reprinted elsewhere in Latin America.

Meantime, a development of natural language processing tools is quite necessary for any country to be competitive in the twenty-first century. We hope that our books will contribute to such developments in Mexico.

CONCLUSIONS

The twenty-first century will be the century of the total information revolution. The development of the tools for the automatic process-

[1] The term *programmed* is sometimes used for such manuals, this term being related to the structure of the manual rather than to a computer program.

ing of the natural language spoken in a country or a whole group of countries is extremely important for the country to be competitive both in science and technology.

To develop such applications, specialists in computer science need to have adequate tools to investigate language with a view to its automatic processing. One of such tools is a deep knowledge of both computational linguistics and general linguistic science.

II. A HISTORICAL OUTLINE

A COURSE ON LINGUISTICS usually follows one of the general models, or theories, of natural language, as well as the corresponding methods of interpretation of the linguistic phenomena.

A comparison with physics is appropriate here once more. For a long time, the Newtonian theory had excluded all other methods of interpretation of phenomena in mechanics. Later, Einstein's theory of relativity incorporated the Newtonian theory as an extreme case, and in its turn for a long time excluded other methods of interpretation of a rather vast class of phenomena. Such exclusivity can be explained by the great power of purely mathematical description of natural phenomena in physics, where theories describe well-known facts and predict with good accuracy the other facts that have not yet been observed.

In general linguistics, the phenomena under investigation are much more complicated and variable from one object (i.e., language) to another than in physics. Therefore, the criteria for accuracy of description and prediction of new facts are not so clear-cut in this field, allowing different approaches to coexist, affect each other, compete, or merge. Because of this, linguistics has a rich history with many different approaches that formed the basis for the current linguistic theories.

Let us give now a short retrospective of the development of general linguistics in the twentieth century. The reader should not be worried if he or she does not know many terms in this review not yet introduced in this book. There will be another place for strict definitions in this book.

THE STRUCTURALIST APPROACH

At the beginning of the twentieth century, *Ferdinand de Saussure* had developed a new theory of language. He considered natural language as a structure of mutually linked elements, similar or opposed to each other. Later, several directions arose in general linguistics and many of them adhered to the same basic ideas about language. This common method was called *structuralism*, and the corresponding scientific research (not only in linguistics, but also in other sciences belonging to the humanities) was called *structuralist*.

Between the 1920's and the 1950's, several structuralist schools were working in parallel. Most of them worked in Europe, and European structuralism kept a significant affinity to the research of the previous periods in its terminology and approaches.

Meantime, American structuralists, Leonard Bloomfield among them, made claims for a fully "objective" description of natural languages, with special attention to superficially observable facts. The order of words in a sentence was considered the main tool to become aware of word grouping and sentence structures. At this period, almost every feature of English seemed to confirm this postulate. The sentences under investigation were split into the so-called immediate constituents, or phrases, then these constituents were in their turn split into subconstituents, etc., down to single words. Such a method of syntactic structuring was called the *phrase structure*, or *constituency*, approach.

INITIAL CONTRIBUTION OF CHOMSKY

In the 1950's, when the computer era began, the eminent American linguist *Noam Chomsky* developed some new formal tools aimed at a better description of facts in various languages [12].

Among the formal tools developed by Chomsky and his followers, the two most important components can be distinguished:

- A purely mathematical nucleus, which includes *generative grammars* arranged in a hierarchy of grammars of diverse complexity. The generative grammars produce strings of symbols, and sets of these strings are called *formal languages*, whereas in general linguistics they could be called texts. Chomskian hierarchy is taught to specialists in computer science, usually in a course on languages and automata. This redeems us from necessity to go into any details. The *context-free* grammars constitute one level of this hierarchy.

- Attempts to describe a number of artificial and natural languages in the framework of generative grammars just mentioned. The phrase structures were formalized as *context-free grammars* (CFG) and became the basic tool for description of natural languages, in the first place, of English. Just examples of these first attempts are extensively elaborated in the manuals on artificial intelligence. It is a good approach unless a student becomes convinced that it is the only possible.

A SIMPLE CONTEXT-FREE GRAMMAR

Let us consider an example of a context-free grammar for generating very simple English sentences. It uses the initial symbol ***S*** of a sentence to be generated and several other *non-terminal* symbols: the noun phrase symbol ***NP***, verb phrase symbol ***VP***, noun symbol ***N***, verb symbol ***V***, and determinant symbol ***D***. All these non-terminal symbols are interpreted as *grammatical categories*.

Several *production rules* for replacement of a non-terminal symbol with a string of several other non-terminal symbols are used as the nucleus of any generative grammar. In our simple case, let the set of the rules be the following:

S → ***NP VP***
VP → ***V NP***
NP → ***D N***
NP → ***N***

Each symbol at the right side of a rule is considered a constituent of the entity symbolized at the left side. Using these rules in any possible order, we can transform ***S*** to the strings ***D N V D N***, or ***D N V N***, or ***N V D N***, or ***N V N***, etc.

An additional set of rules is taken to convert all these non-terminal symbols to the *terminal* symbols corresponding to the given grammatical categories. The terminals are usual words of Spanish, English, or any other language admitting the same categories and the same word order. We use the symbol "|" as a metasymbol of an alternative (i.e. for logical *OR*). Let the rules be the following:

N → *estudiante* | *niña* | *María* | *canción* | *edificio*...
V → *ve* | *canta* | *pregunta*...
D → *el* | *la* | *una* | *mi* | *nuestro*...

Applying these rules to the constituents of the non-terminal strings obtained earlier, we can construct a lot of fully grammatical and meaningful Spanish sentences like *María ve el edificio* (from ***N V D N***) or *la estudiante canta una canción* (from ***D N V D N***). Some meaningless and/or ungrammatical sentences like *canción ve el María* can be generated too. With more complicate rules, some types of ungrammaticality can be eliminated. However, to fully get rid of potentially meaningless sentences is very difficult, since from the very beginning the initial symbol does not contain any specific meaning at all. It merely presents an abstract category of a sentence of a very vast class, and the resulting meaning (or nonsense) is accumulated systematically, with the development of each constituent.

On the initial stage of the elaboration of the generative approach, the idea of independent syntax arose and the problem of natural language processing was seen as determining the *syntactic structure* of each sentence composing a text. Syntactic structure of a sentence was identified with the so-called *constituency tree*. In other words, this is a nested structure subdividing the sentence into parts, then these parts into smaller parts, and so on. This decomposition corresponds to the sequence of the grammar rules applications that gen-

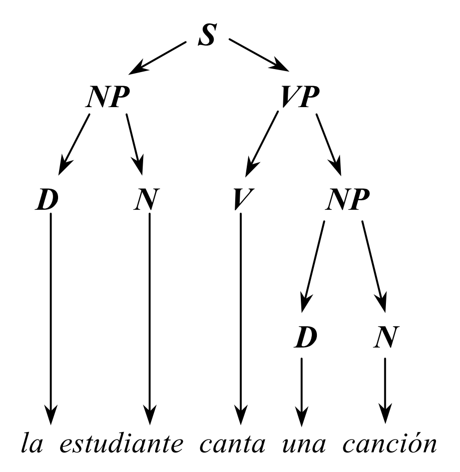


FIGURE II.1. *Example of constituency tree.*

erate the given sentence. For example, the Spanish sentence *la estudiante canta una canción* has the constituency tree represented graphically in Figure II.1. It also can be represented in the form of the following nested structure marked with square brackets:

$$\left[\left[[\textit{la}]_{D}\ [\textit{estudiante}]_{N}\right]_{NP}\ \left[[\textit{canta}]_{V}\ \left[[\textit{una}]_{D}\ [\textit{canción}]_{N}\right]_{NP}\right]_{VP}\right]_{S}$$

This structure shows the sentence ***S*** consisting of a noun phrase ***NP*** and a verb phrase ***VP***, that in its turn consists of a verb ***V*** followed by a noun phrase ***NP***, that in its turn consists of a determiner ***D*** (an article or pronoun) followed by a noun ***N*** that is the word *canción*, in this case.

TRANSFORMATIONAL GRAMMARS

Further research revealed great generality, mathematical elegance, and wide applicability of generative grammars. They became used not only for description of natural languages, but also for specification of formal languages, such as those used in mathematical logic,

pattern recognition, and programming languages. A new branch of science called *mathematical linguistics* (in its narrow meaning) arose from these studies.

During the next three decades after the rise of mathematical linguistics, much effort was devoted to improve its tools for it to better correspond to facts of natural languages. At the beginning, this research stemmed from the basic ideas of Chomsky and was very close to them.

However, it soon became evident that the direct application of simple context-free grammars to the description of natural languages encounters great difficulties. Under the pressure of purely linguistic facts and with the aim to better accommodate the formal tools to natural languages, Chomsky proposed the so-called *transformational grammars*. They were mainly English-oriented and explained how to construct an interrogative or negative sentence from the corresponding affirmative one, how to transform the sentence in active voice to its passive voice equivalent, etc.

For example, an interrogative sentence such as *Does John see Mary?* does not allow a nested representation as the one shown on page 37 since the two words *does* and *see* obviously form a single entity to which the word *John* does not belong. Chomsky's proposal for the description of its structure consisted in

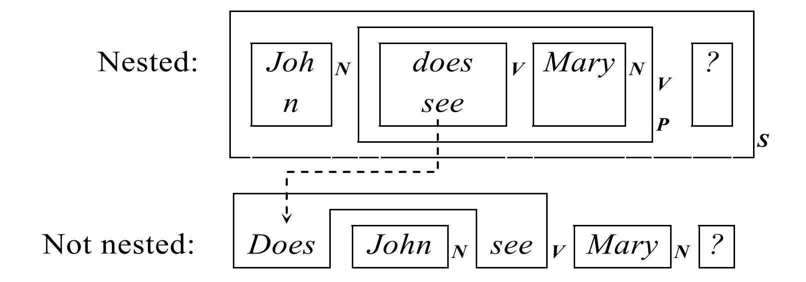
(a) description of the structure of some "normal" sentence that does permit the nested representation plus
(b) description of a process of obtaining the sentence in question from such a "normal" sentence by its *transformation*.

Namely, to construct the interrogative sentence from a "normal" sentence "*John sees Mary.*", it is necessary

(1) to replace the period with the question mark (**John sees Mary?*),
(2) to transform the personal verb form *see* into a word combination *does see* (**John does see Mary?*), and finally

(3) to move the word *does* to the beginning of the sentence (*Does John see Mary?*), the latter operation leading to the "violation" of the nested structure.

This is shown in the following figure:



A transformational grammar is a set of rules for such insertions, permutations, movements, and corresponding grammatical changes. Such a set of transformational rules functions like a program. It takes as its input a string constructed according to some context-free grammar and produces a transformed string.

The application of transformational grammars to various phenomena of natural languages proved to be rather complicated. The theory has lost its mathematical elegance, though it did not acquire much of additional explanatory capacity.

THE LINGUISTIC RESEARCH AFTER CHOMSKY: VALENCIES AND INTERPRETATION

After the introduction of the Chomskian transformations, many conceptions of language well known in general linguistics still stayed unclear. In the 1980's, several grammatical theories different from Chomsky's one were developed within the same phrase-structure mainstream. Nearly all of them were based on the CFGs, but used different methods for description of some linguistic facts.

One very important linguistic idea had been suggested already by Chomsky and adopted by the newer theories. It is the subcategorization of verbs according to their ability to accept specific sets of

complements. These complements are also called *actants*, or *valency fillers,* which we will use interchangeably. We also will informally use the term *valency* for *valency filler*, though valency is a link, whereas a valency filler is a linked word or a word group.

The term *valency* is also used in chemistry, and this is not by accident. The point is that each specific verb has its own standard set of actants (usually nouns). Within a sentence, the actants are usually located close to the predicative verb and are related with it semantically. For example, the Spanish verb *dar* has three actants reflecting (1) donator (who gives?), (2) donation (what is given?) and (3) receptor (to whom is given?).

In texts, the valencies are given in specific ways depending on the verb, e.g., with a specific word order and/or a specific preposition for each valency. All three actants of the verb *dar* can be seen in the sentence *Juan* (1) *dio muchas flores* (2) *a Elena* (3). The last two actants and their position in a sentence after the verb *dar* can be expressed with the pattern:

dar <donation> *a* <receptor>

The description given above reflects linguistic phenomena including both syntactic and semantic aspects of valencies. In particular, the names <donator>, <donation>, and <receptor> reflect valencies in their semantic aspect. As to the generative grammar approach, it operates only with constituents and related grammar categories. Under this approach, the pattern called *subcategorization frame* has the form:

dar $\boldsymbol{N}_1$ *a* $\boldsymbol{N}_2$,

where $\boldsymbol{N}_1$ and $\boldsymbol{N}_2$ stand for noun phrases, without exposure of their semantic roles. Thus, these phrases are not distinguishable semantically, they only can be given different syntactic interpretations: $\boldsymbol{N}_1$ is a direct complement and $\boldsymbol{N}_2$ is an indirect complement.

We have induced only one of the possible subcategorization frames for the verb *dar*. To reflect the structure of the sentence *Juan* (1) *dio a Elena* (3) *muchas flores* (2), with the same semantic valen-

cies given in a different order, we are compelled to introduce another pattern:

dar a <receptor> <donation>

with the corresponding subcategorization frame

dar a $\boldsymbol{N}_1$ $\boldsymbol{N}_2$.

Direct and indirect complements swap over, while their semantic roles stay the same, bit it is not clear in such a subcategorization frame.

Categorization and subcategorization are kinds of classification. Any consistent classification implies separation of entities to several non-intersecting groups. However, in the case under our study, the verb *dar* should be considered belonging to two different subcategories. Or else two verbs *dar* should be introduced, with equal meanings and equal semantic valency sets, but with different subcategorization frames. In fact, the situation in languages with the free word order is even more complicated. Indeed, the verb *dar* can have their donation and receptor actants staying before the subject, like in the sentence *A Elena* (3) *le dio Juan* (1) *muchas flores* (2). Such an order requires even more subcategorization frames obligatorily including the subject of the sentence, i.e. the first valency or the verb, with the role of donator.

The actants are used with verbs more frequently than the so-called circonstants. The circonstants are expressed by adverbs or, similarly to actants, by prepositional groups, i.e., through a combination of a preposition and (usually) a noun. However, the way they are expressed in text does not usually depend on a specific verb. Thus, in the first approximation, the difference between actants and circonstants can be roughly summarized as follows.

- In the syntactic aspect, actants are expressed peculiarly depending on the specific verb, whereas circonstants do not depend on the specific verb in their form, and
- In the semantic aspect, actants are obligatory participants of the situation described by the verb, while circonstants are not.

Only one, obligatory and usually the most important, participant of the situation is expressed by many languages in a quite standard form, namely the subject of the sentence. In Spanish, English, and several other languages (but not in all of them!), it usually precedes the *syntactic predicate* of the sentence and is represented with a noun without any preposition. Since the subject is expressed in the same manner with all verbs, it is not specified explicitly in the subcategorization frames. However, it is efficient only for languages with strict word order.

As we could see above, the semantic interpretation of different *phrases* within a sentence cannot be given in the frame of the purely generative approach. It can only distinguish which noun phrase within a sentence is subject, or direct complement, or indirect complement, etc. In deep semantic interpretation ("understanding"), additional theoretical means were needed, and they were first found out of the generative approach. In late 60s, Charles Fillmore [13] has introduced *semantic valencies* under the name of *semantic cases*. Each verb has its own set of semantic cases, and the whole set of verbs in any language supposedly has a finite and rather limited inventory of all possible semantic cases. Just among them, we can see semantic cases of donator, donation, and receptor sufficient for interpretation of the verb *dar*. To "understand" any verb deeper, some rules connecting subcategorization frames and semantic cases had been introduced.

LINGUISTIC RESEARCH AFTER CHOMSKY: CONSTRAINTS

Another very valuable idea originated within the generative approach was that of using special features assigned to the constituents, and specifying *constraints* to characterize agreement or coordination of their grammatical properties. For example, the rule ***NP*** → ***D N*** in Spanish and some other languages with morphologic agreement of determinants and the corresponding nouns incorrectly admits constituents like **unas libro*. To filter out such incorrect

combinations, this rule can be specified in a form similar to an equation:

$$\boldsymbol{NP}(\text{Gen, Num}) \rightarrow \boldsymbol{D}(\text{Gen, Num})\ \boldsymbol{N}(\text{Gen, Num}),$$

where Gen and Num are variables standing for any specific gender and any specific number, correspondingly. The gender Gen and the number Num of a determinant should be the same as (i.e., agree with) the gender Gen and the number Num of the consequent noun (compare Spanish *la luna* and *el sol*, but not **unas libro*). Such a notation is shorthand of the following more complete and more frequently used notation:

$$\boldsymbol{NP}\begin{bmatrix}\text{gender}: & \text{Gen}\\ \text{number}: & \text{Num}\end{bmatrix} \rightarrow \boldsymbol{D}\begin{bmatrix}\text{gender}: & \text{Gen}\\ \text{number}: & \text{Num}\end{bmatrix} \boldsymbol{N}\begin{bmatrix}\text{gender}: & \text{Gen}\\ \text{number}: & \text{Num}\end{bmatrix}.$$

The following variant is also used, where the same value is not repeated, but is instead assigned a number and then referred to by this number. Thus, $\boxed{1}$ stands for the value Gen and $\boxed{2}$ for the value Num as specified where these numbers first occur:

$$\boldsymbol{NP}\begin{bmatrix}\text{gender}: & \boxed{1}\text{Gen}\\ \text{number}: & \boxed{2}\text{Num}\end{bmatrix} \rightarrow \boldsymbol{D}\begin{bmatrix}\text{gender}: & \boxed{1}\\ \text{number}: & \boxed{2}\end{bmatrix} \boldsymbol{N}\begin{bmatrix}\text{gender}: & \boxed{1}\\ \text{number}: & \boxed{2}\end{bmatrix}.$$

The same constraint can be expressed in an even shorter equivalent form, where $\boxed{1}$ stands for the whole combination of the two features as specified when the symbol $\boxed{1}$ first occurs:

$$\boldsymbol{NP}\ \boxed{1}\begin{bmatrix}\text{gender}: & \text{Gen}\\ \text{number}: & \text{Num}\end{bmatrix} \rightarrow \boldsymbol{D}\boxed{1}\ \boldsymbol{N}\boxed{1}.$$

Each feature of a constituent can be expressed with a pair: its name and then its value, e.g., “gender: Gen”. The rule above determines which values are to be equal. With each constituent, any number of features can be expressed, while different constituents within the same rule possibly can have different sets of specified features.

Another example of the correspondence between features of constituents is the following rule of agreement in person and number between the subject and the predicate of a Spanish sentence (the syntactic predicate is a verb in finite, i.e., in personal form):

S → ***NP***(Pers, Num) ***VP***(Pers, Num).

It means that, in the Spanish phrase like *yo quiero* or *ellas quieren*, there is a correspondence between the values of grammatical persons Pers and numbers Num: in the first phrase, Pers = 1^{st}, Num = singular, while in the second phrase, Pers = 3^{rd}, Num = plural on the both sides.

The constraint-based approach using the features was intensively used by the *Generalized Phrase Structure Grammar* (GPSG), and now is generally accepted. The featured notation permits to diminish the total number of rules. A generalized rule covers several simple rules at once. Such approach paves way to the method of unification to be exposed later.

HEAD-DRIVEN PHRASE STRUCTURE GRAMMAR

One of the direct followers of the GPSG was called Head-Driven Phrase Structure Grammar (HPSG). In addition to the advanced traits of the GPSG, it has introduced and intensively used the notion of *head*. In most of the constituents, one of the sub-constituents (called *daughters* in HPSG) is considered as the principal, or its head (called also *head daughter*). For example, in the rule:

S → ***NP***(Pers, Num) H***VP***(Pers, Num),

the ***VP*** constituent (i.e., the syntactic predicate of a sentence with all connected words) is marked as the head of the whole sentence, which is indicated by the symbol H. Another example: in the rule:

NP (Gen, Num) → ***D*** (Gen, Num) H***N*** (Gen, Num),

the noun is marked as the head of the whole noun phrase ***NP***.

According to one of the special *principles* introduced in HPSG, namely the head principle, the main features of the head are inherited in some way by the mother (enclosing) constituent (the left-side part of the rule).

In the previous examples, the features of the predicate determine features of the whole sentence, and the features of the noun determine the corresponding features of the whole noun phrase. Such formalism permits to easier specify the syntactic structure of sentences and thus facilitates syntactic analysis (parsing).

As it was already said, the interpretation in early generative grammars was always of syntactic nature. For semantic interpretation ("understanding"), additional theoretical means were introduced, which were somewhat alien to the earlier generative structure mainstream. By contrast, each word in the HPSG dictionary is supplied with semantic information that permits to combine meanings of separate words into a joint coherent semantic structure. The novel rules of the word combining gave a more adequate method of construing the semantic networks. Meantime, Chomskian idea of transformations was definitely abandoned by this approach.

THE IDEA OF UNIFICATION

Having in essence the same initial idea of phrase structures and their context-free combining, the HPSG and several other new approaches within Chomskian mainstream select the general and very powerful mathematical conception of *unification.* The purpose of unification is to make easier the syntactic analysis of natural languages.

The unification algorithms are not linguistic proper. Rather they detect similarities between parts of mathematical structures (strings, trees, graphs, logical formulas) labeled with feature sets. A priori, it is known that some features are interrelated, i.e., they can be equal, or one of them covers the other. Thus, some feature combinations are considered compatible while met in analysis, whereas the rest are not. Two sets of features can be unified, if they are compatible.

Then the information at an object admitting unification (i.e., at a constituent within a sentence to be parsed) combines the information brought by both sets of features.

Unification allows filtering out inappropriate feature options, while the unified feature combination characterizes the syntactic structure under analysis more precisely, leading to the true interpretation of the sentence.

As the first example of unification operations, let us compare feature sets of two Spanish words, *el* and *muchacho,* staying in a text side by side. Both words have the feature set [gender = masculine, number = singular], so that they are equivalent with respect to gender and number. Hence, the condition of unification is satisfied, and this pair of words can form a unifying constituent in syntactic analysis.

Another example is the adjacent Spanish words *las estudiantes*. The article *las* has the feature set [gender = feminine, number = plural]. As to the string *estudiantes*, this word can refer to both 'he-student' of masculine gender and 'she-student' of feminine gender, so that this word is not specified (is *underspecified*) with respect to gender. Thus, the word occurrence *estudiantes* taken separately has a broader feature set, namely, [number = plural], without any explicit indication of gender. Since these two feature sets are not contradictory, they are compatible and their unification [gender = feminine, number = plural] gives the unifying constraint set assigned to both words. Hence, this pair can form a unifying mother constituent *las estudiantes,* which inherits the feature set from the head daughter *estudiantes*. The gender of the particular word occurrence *estudiantes* is feminine, i.e., 'she-students,' and consequently the inherited gender of the noun phrase *las estudiantes* is also feminine.

As the third example, let us consider the words *niño* and *quisiera* in the Spanish sentence *El niño quisiera pasar de año*. The noun *niño* is labeled with the 3rd person value: [person = 3], whereas the verb *quisiera* exists in two variants labeled with the feature set [person = 1 or person = 3], correspondingly. Only the latter variant of

the verb can be unified with the word *niño*. Therefore this particular word occurrence of *quisiera* is of the third person. The whole sentence inherits this value, since the verb is its head daughter.

THE MEANING ⇔ TEXT THEORY: MULTISTAGE TRANSFORMER AND GOVERNMENT PATTERNS

The European linguists went their own way, sometimes pointing out some oversimplifications and inadequacies of the early Chomskian linguistics.

In late 1960´s, a new theory, the Meaning ⇔ Text model of natural languages, was suggested in Russia. For more than 30 years, this theory has been developed by the scientific teams headed by I. Mel'čuk, in Russia and then in Canada, and by the team headed by Yu. Apresian in Russia, as well as by other researchers in various countries. In the framework of the Meaning ⇔ Text Theory (MTT), deep and consistent descriptions of several languages of different families, Russian, French, English and German among them, were constructed and introduced to computational practice.

One very important feature of the MTT is considering the language as multistage, or multilevel, transformer of meaning to text and vice versa. The transformations are comprehended in a different way from the theory by Chomsky. Some inner representation corresponds to each level, and each representation is equivalent to representations of other levels. Namely, surface morphologic, deep morphologic, surface syntactic, deep syntactic, and semantic levels, as well as the corresponding representations, were introduced into the model.

The description of valencies for words of any part of speech and of correspondence between the semantic and syntactic valencies have found their adequate solution in this theory, in terms of the so-called *government patterns*.

The government patterns were introduced not only for verbs, but also for other parts of speech. For a verb, GP has the shape of a table

of all its possible valency representations. The table is preceded by the formula of the semantic interpretation of the situation reflected by the verb with all its valencies. The table is succeeded by information of word order of the verb and its actants.

If to ignore complexities implied by Spanish pronominal clitics like *me*, *te*, *se*, *nos*, etc., the government pattern of the Spanish verb *dar* can be represented as

Person X gives thing Y to person Z

X = 1	Y = 2	Z = 3
1.1 ***N***	2.1 ***N***	3.1 *a* ***N***

The symbols X, Y, and Z designate semantic valencies, while 1, 2, and 3 designate the syntactic valencies of the verb. Meaning ‘give’ in the semantic formula is considered just corresponding to the Spanish verb *dar*, since *dar* supposedly cannot be represented by the more simple semantic elements.

The upper line of the table settles correspondences between semantic and syntactic valencies. For this verb, the correspondence is quite simple, but it is not so in general case.

The lower part of the table enumerates all possible options of representation for each syntactic valency at the syntactic level. The options operate with part-of-speech labels (***N*** for a noun, $\boldsymbol{V}_{\mathrm{inf}}$ for a verb in infinitive, etc.) and prepositions connecting the verb with given valency fillers. In our simplistic example, only nouns can fill all three valencies, only the preposition *a* is used, and each valency have the unique representation. However, such options can be multiple for other verbs in Spanish and various other languages. For example, the English verb *give* has two possible syntactic options for the third valency: without preposition (*John gives* ***him*** *a book*) vs. with the preposition *to* (*John gives a book* ***to him***).

The word order is generally not related with the numeration of the syntactic valencies in a government pattern. If all permutations of valencies of a specific verb are permitted in the language, then no

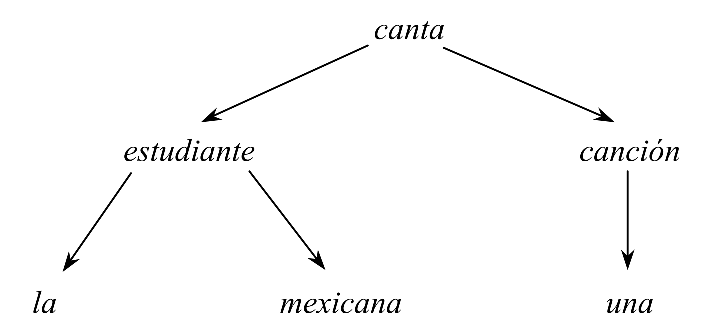


FIGURE II.2. *Example of a dependency tree.*

information about word order is needed in this GP. Elsewhere, information about forbidden or permitted combinations is given explicitly, to make easier the syntactic analysis. For example, the English verb *give* permits only two word orders mentioned above.

Thus, government patterns are all-sufficient for language description and significantly differ from subcategorization frames introduced in the generative grammar mainstream.

THE MEANING ⇔ TEXT THEORY: DEPENDENCY TREES

Another important feature of the MTT is the use of its *dependency trees,* for description of syntactic links between words in a sentence. Just the set of these links forms the representation of a sentence at the syntactic level within this approach.

For example, the Spanish sentence *La estudiante mexicana canta una canción* can be represented by the dependency tree shown in Figure II.2. One can see that the dependency tree significantly differs from the constituency tree for the same sentence (cf. Figure II.1).

Up to the present, the proper description of the word order and word agreement in many languages including Spanish can be accomplished easier by means of the MTT. Moreover, it was shown that in many languages there exist disrupt and non-projective con-

structions, which cannot be represented through constituency trees or nested structures, but dependency trees can represent them easily.

In fact, dependency trees appeared as an object of linguistic research in the works of Lucien Tesnière, in 1950's. Even earlier, dependencies between words were informally used in descriptions of various languages, including Spanish. However, just the MTT has given strict definition to dependency trees. The dependency links were classified for surface and deep syntactic levels separately. They were also theoretically isolated from links of morphologic inter-word agreement so important for Spanish.

With dependency trees, descriptions of the relationships between the words constituting a sentence and of the order of these words in the sentence were separated from each other. Thus, the links between words and the order in which they appear in a sentence were proposed to be investigated apart, and relevant problems of both analysis and synthesis are solved now separately.

Hence, the MTT in its syntactic aspect can be called *dependency approach*, as contrasted to the *constituency approach* overviewed above. In the dependency approach, there is no problem for representing the structure of English interrogative sentences (cf. page 39). Thus, there is no necessity in the transformations of Chomskian type.

To barely characterize the MTT as a kind of dependency approach is to extremely simplify the whole picture. Nevertheless, this book presents the information permitting to conceive other aspects of the MTT.

THE MEANING ⇔ TEXT THEORY: SEMANTIC LINKS

The dependency approach is not exclusively syntactic. The links between wordforms at the surface syntactic level determine links between corresponding labeled nodes at the deep syntactic level, and after some deletions, insertions, and inversions imply links in the semantic representation of the same sentence or a set of sen-

tences. Hence, this approach facilitates the transfer from syntactic representations to a semantic one and vice versa.

According to the MTT, the correlation between syntactic and semantic links is not always straightforward. For example, some auxiliary words in a sentence (e.g., auxiliary verbs and some prepositions) are treated as *surface elements* and disappear at the deep syntactic level. For example, the auxiliary Spanish verb *HABER* in the word combination *han pedido* disappears from the semantic representation after having been used to determine the verb tense and mode. At the same time, some elements absent in the surface representation are *deduced,* or *restored,* from the context and thus appear explicitly at the deep syntactic level. For example, given the surface syntactic dependency tree fragment:

su ← *hijo* → *Juan*,

the semantically conditioned element NAME is inserted at the deep syntactic level, directly ruling the personal name:

su ← *hijo* → NAME → *Juan*

Special rules of inter-level correspondence facilitate the transition to the correct semantic representation of the same fragment.

The MTT provides also the rules of transformation of some words and word combinations to other words and combinations, with the full preservation of the meaning. For example, the Spanish sentence *Juan me prestó ayuda* can be formally transformed to *Juan me ayudó* and vice versa at the deep syntactic level. Such transformations are independent of those possible on the semantic level, where mathematical logic additionally gives quite other rules of meaning-preserving operations.

We should clarify that some terms, e.g., *deep structure* or *transformation*, are by accident used in both the generative and the MTT tradition, but in completely different meanings. Later we will return to this source of confusion.

All these features will be explained in detail later. Now it is important for us only to claim that the MTT has been able to describe any natural language and any linguistic level in it.

CONCLUSIONS

In the twentieth century, syntax was in the center of the linguistic research, and the approach to syntactic issues determined the structure of any linguistic theory. There are two major approaches to syntax: the constituency, or phrase-structure, approach, and the dependency approach. The constituency tradition was originated by N. Chomsky with the introduction of the context-free grammars, and the most recent development in this tradition is Head-driven Phrase Structure Grammar theory. The dependency approach is used in the Meaning ⇔ Text Theory by Igor Mel'čuk. Both approaches are applicable for describing linguistic phenomena in many languages.

III. PRODUCTS OF COMPUTATIONAL LINGUISTICS: PRESENT AND PROSPECTIVE

FOR WHAT PURPOSES do we need to develop computational linguistics? What practical results does it provide for society? Before we start discus-sing the methods and techniques of computational linguistics, it is worthwhile giving a review of some existing practical results, i.e., applications, or products, of this discipline. We consider such applications in a very broad sense, including in this category all known tasks of word processing, as well as those of text processing, text generation, dialogue in a natural language, and language understanding.

Some of these applications already provide the user with satisfactory solutions for their tasks, especially for English, while other tasks and languages have been under continuous research in recent decades.

Of course, some extrapolations of the current trends could give completely new types of systems and new solutions to the current problems, but this is out of scope of this book.

CLASSIFICATION OF APPLIED LINGUISTIC SYSTEMS

Applied linguistic systems are now widely used in business and scientific domains for many purposes. Some of the most important ones among them are the following:

- *Text preparation*, or text editing, in a broad sense, particularly including the tasks listed below:
  - *Automatic hyphenation* of words in natural language texts,
  - *Spell checking*, i.e., detection and correction of typographic and spelling errors,

  – *Grammar checking*, i. e., detection and correction of grammatical errors,
  – *Style checking*, i. e. detection and correction of stylistic errors,
  – *Referencing* specific words, word combinations, and semantic links between them;

- *Information retrieval* in scientific, technical, and business document databases;
- *Automatic translation* from one natural language to another;
- *Natural language interfaces* to databases and other systems;
- *Extraction of factual data* from business or scientific texts;
- *Text generation* from pictures and formal specifications;
- *Natural language understanding*;
- *Optical character recognition*, *speech recognition*, etc.

For the purposes of this book, we will give here only a short sketch of each application. Later, some of these topics, with more deep explanations, can be touched upon once more.

AUTOMATIC HYPHENATION

Hyphenation is intended for the proper splitting of words in natural language texts. When a word occurring at the end of a line is too long to fit on that line within the accepted margins, a part of it is moved to the next line. The word is thus wrapped, i.e., split and partially transferred to the next line.

The wrapping can be done only at specific positions within words, which generally, though not always, are syllable boundaries. For example, in Spanish one can split *re-ci-bo, re-u-nir-se, dia-blo, ca-rre-te-ra, mu-cha-chas*, but not in the following positions: **recib-o, *di-ablo, *car-retera, *muc-hac-has.*

In this way, hyphenation improves the outer appearance of computer-produced texts through adjusting their right margins. It saves paper and at the same time preserves impression of smooth reading, just as without any hyphenation.

The majority of the well-known text editors are supplied now with hyphenation tools. For example, Microsoft Word has the menu item *Hyphenation*.[2]

Usually, the linguistic information taken for such programs is rather limited. It should be known which letters are *vowels* (*a, e, i, o, u* in Spanish) or *consonants* (*b, c, d, f, g,* etc.), and what letter combinations are inseparable (such as consonants pairs *ll*, *rr*, *ch* or diphthongs *io, ue, ai* in Spanish).

However, the best quality of hyphenation could require more detailed information about each word. The hyphenation can depend on the so-called morphemic structure of the word, for example: *sub-ur-ba-no*, but *su-bir*, or even on the origin of the word, for example: *Pe-llicer*, but *Shil-ler*. Only a dictionary-based program can take into account all such considerations. For English, just dictionary-based programs really give perfect results, while for Spanish rather simple programs are usually sufficient, if to neglect potentially error-prone foreign words like *Shiller*.

SPELL CHECKING

The objective of spell checking is the detection and correction of typographic and orthographic errors in the text at the level of *word occurrence* considered out of its context.

Nobody can write without any errors. Even people well acquainted with the rules of language can, just by accident, press a wrong key on the keyboard (maybe adjacent to the correct one) or miss out a letter. Additionally, when typing, one sometimes does not synchronize properly the movements of the hands and fingers. All such errors are called *typos*, or *typographic errors*. On the other

[2] *Guiones* in the Spanish version.

hand, some people do not know the correct spelling of some words, especially in a foreign language. Such errors are called *spelling errors*.

First, a spell checker merely detects the strings that are not correct words in a given natural language. It is supposed that most of the orthographic or typographic errors lead to strings that are *impossible* as separate words in this language. Detecting the errors that convert by accident one word into another existing word, such as English *then* → ?*than* or Spanish *cazar* → ?*casar*, supposes a task which requires much more powerful tools.

After such impossible string has been detected and highlighted by the program, the user can correct this string in any preferable way—manually or with the help of the program. For example, if we try to insert into any English text the strings[3] **groop*, **greit*, or **misanderstand*, the spell checker will detect the error and stop at this string, highlighting it for the user. Analogous examples in Spanish can be **caió*, **systema*, **nesecitar*.

The functions of a spell checker can be more versatile. The program can also propose a set of existing words, which are *similar* enough (in some sense) to the given corrupted word, and the user can then choose one of them as the correct version of the word, without re-typing it in the line. In the previous examples, Microsoft Word's spell checker gives, as possible candidates for replacement of the string *caió*, the existing Spanish words shown in Figure III.1.

In most cases, especially for long strings, a spell checker offers only one or two candidates (or none). For example, for the string **systema* it offers only the correct Spanish word *sistema*.

The programs that perform operations of both kinds are called orthographic correctors, while in English they are usually called spell checkers. In everyday practice, spell checkers are considered very helpful and are used by millions of users throughout the world. The majority of modern text editors are supplied now with integrated

[3] In all cases, an initial asterisk marks strings containing errors.



FIGURE III.1. *Alternatives for the word* *caió.

spell checkers. For example, Microsoft Word uses many spell checkers, a specific one for each natural language used in the text.

The amount of linguistic information necessary for spell checkers is much greater than for hyphenation. A simple but very resource-consuming approach operates with a list, or a dictionary, of all valid words in a specific language. It is necessary to have also a criterion of similarity of words, and some presuppositions about the most common typographic and spelling errors. A deeper penetration into the correction problems requires a detailed knowledge of morphology, since it facilitates the creation of a more compact dictionary that has a manageable size.

Spell checkers have been available for more than 20 years, but some quite evident tasks of correction of words, even taken separately, have not been yet solved. To put a specific example, let us consider the ungrammatical string **teached* in an English text. None of the spell checkers we have tried suggested the correct form *taught*. In an analogous way, if a foreigner inserts into a Spanish text such strings as **muestrar* or **disponido*, the Spanish spell checkers we have tried did not give the forms *mostrar* and *dispuesto* as possible corrections.

GRAMMAR CHECKING

Detection and correction of grammatical errors by taking into account adjacent words in the sentence or even the whole sentence are much more difficult tasks for computational linguists and software developers than just checking orthography.

Grammar errors are those violating, for example, the syntactic laws or the laws related to the structure of a sentence. In Spanish, one of these laws is the agreement between a noun and an adjective in gender and grammatical number. For example, in the combination **mujer viejos* each word by itself does exist in Spanish, but together they form a syntactically ill-formed combination. Another example of a syntactic agreement is the agreement between the noun in the role of subject and the main verb, in number and person (**tú tiene*).

The words that must agree can be located in quite different parts of the sentence. For example, it is rather difficult for a program to find the error in the following sentence: **Las mesas de madera son muy largos*.

Other types of grammatical errors include incorrect usage of prepositions, like in the phrases **debajo la puerta*, or **¡basta con verla!*, or **casarse a María*. Some types of syntactic errors may be not so evident even for a native speaker.

It became clear long ago that only a complete syntactic analysis (parsing) of a text could provide an acceptable solution of this task. Because of the difficulty of such parsing, commercial grammar checkers are still rather primitive and rarely give the user useful assistance in the preparation of a text. The *Windows Sources*, one of the well-known computer journals, noted, in May 1995, that the grammar checker Grammatik in the WordPerfect text editor, perhaps the best grammar checker in the world at that time, was so imperfect and disorienting, that “nobody needs a program that’s wrong in more cases than it’s right.”

In the last few years, significant improvements have been made in grammar checkers. For example, the grammar checker included in

Microsoft Word is helpful but still very far from perfection.

Sometimes, rather simple operations can give helpful results by detecting some very frequent errors. The following two classes of errors specific for Spanish language can be mentioned here:

- Absence of agreement between an article and the succeeding noun, in number and gender, like in **la gatos*. Such errors are easily detectable within a very narrow context, i.e., of two adjacent words. For this task, it is necessary to resort to the grammatical categories for Spanish words.

- Omission of the written accent in such nouns as **articulo,* **genero,* **termino.* Such errors cannot be detected by a usual spell checker taking the words out of context, since they convert one existing word to another existent one, namely, to a personal form of a verb. It is rather easy to define some properties of immediate contexts for nouns that never occur with the corresponding verbs, e.g., the presence of agreed articles, adjectives, or pronouns [38].

We can see, however, that such simplistic techniques fail in too many cases. For example, in combinations such as **las pruebas de evaluación numerosos,* the disagreement between *pruebas* and *numerosos* cannot be detected by considering only the nearest context.

What is worse, a program based on such a simplistic approach would too frequently give false alarms where there is no error in fact. For example, in the correct combination *las pruebas de evaluación numerosas*, such a simplistic program would mention disagreement in number between the wordforms *evaluación* and *numerosas*.

In any case, since the author of the text is the only person that definitely knows what he or she meant to write, the final decision must always be left up to the user, whether to make a correction suggested by the grammar checker or to leave the text as it was.

STYLE CHECKING

The stylistic errors are those violating the laws of use of correct words and word combinations in language, in general or in a given literary genre.

This application is the nearest in its tasks to normative grammars and manuals on stylistics in the printed, oriented to humans, form. Thus, style checkers play a didactic and prescriptive role for authors of texts.

For example, you are not recommended to use any vulgar words or purely colloquial constructions in official documents. As to more formal properties of Spanish texts, their sentences should not normally contain ten prepositions *de*, and should not be longer than, let us say, twenty lines. With respect to Spanish lexicon, it is not recommended to use the English words *parking* and *lobby* instead of *estacionamiento* and *vestíbulo*, or to use the Americanism *salvar* in the meaning 'to save in memory' instead of *guardar*.

In the Spanish sentence *La recolección de datos en tiempo* ***real es realizada*** *mediante un servidor*, the words in boldface contain two stylistic anomalies: *se realiza* is usually better than *es realizada,* and such a close neighborhood of words with the same stem, like *real* and *realizada*, is unwanted.

In the Spanish sentence *La* ***grabación****,* ***reproducción*** *y* ***simulación*** *de datos son funciones en todos los sistemas de* ***manipulación*** *de* ***información***, the frequency of words with the suffix -*ción* oversteps limits of a good style.

The style checker should use a dictionary of words supplied with their usage marks, synonyms, information on proper use of prepositions, compatibility with other words, etc. It should also use automatic parsing, which can detect improper syntactic constructions.

There exist style checkers for English and some other major languages, but mainly in laboratory versions. Meanwhile commercial style checkers are usually rather primitive in their functions.

As a very primitive way to assess stylistic properties of a text, some commercial style checkers calculate the average length of

words in the text, i.e., the number of letters in them; length of sentences, i.e., the number of words in them; length of paragraphs, i.e., the number of words and sentences. They can also use other statistical characteristics that can be easily calculated as a combination of those mentioned.

The larger the average length of a word, sentence or paragraph, the more difficult the text is to read, according to those simplest stylistic assessments. It is easy also to count the occurrences of prepositions *de* or nouns ending in -*ción* in Spanish sentences.

Such style checkers can only tell the user that the text is too complicated (awkward) for the chosen genre, but usually cannot give any specific suggestions as to how to improve the text.

The assessment of deeper and more interesting stylistic properties, connected with the lexicon and the syntactic constructions, is still considered a task for the future.

REFERENCES TO WORDS AND WORD COMBINATIONS

The references from any specific word give access to the set of words semantically related to the former, or to words, which can form combinations with the former in a text. This is a very important application. Nowadays it is performed with linguistic tools of two different kinds: autonomous on-line dictionaries and built-in dictionaries of synonyms.

Within typical text processors, the synonymy dictionaries are usually called thesauri. Later we will see that this name corresponds poorly to the synonymy dictionaries, since genuine thesauri usually include much more information, for example, references to generic words, i.e., names of superclasses, and to specific words, i.e., names of subclasses.

References to various words or word combinations of a given natural language have the objective to help the author of a text to create more correct, flexible, and idiomatic texts. Indeed, only an insignificant part of all thinkable word combinations are really permitted in a language, so that the knowledge of the permitted and

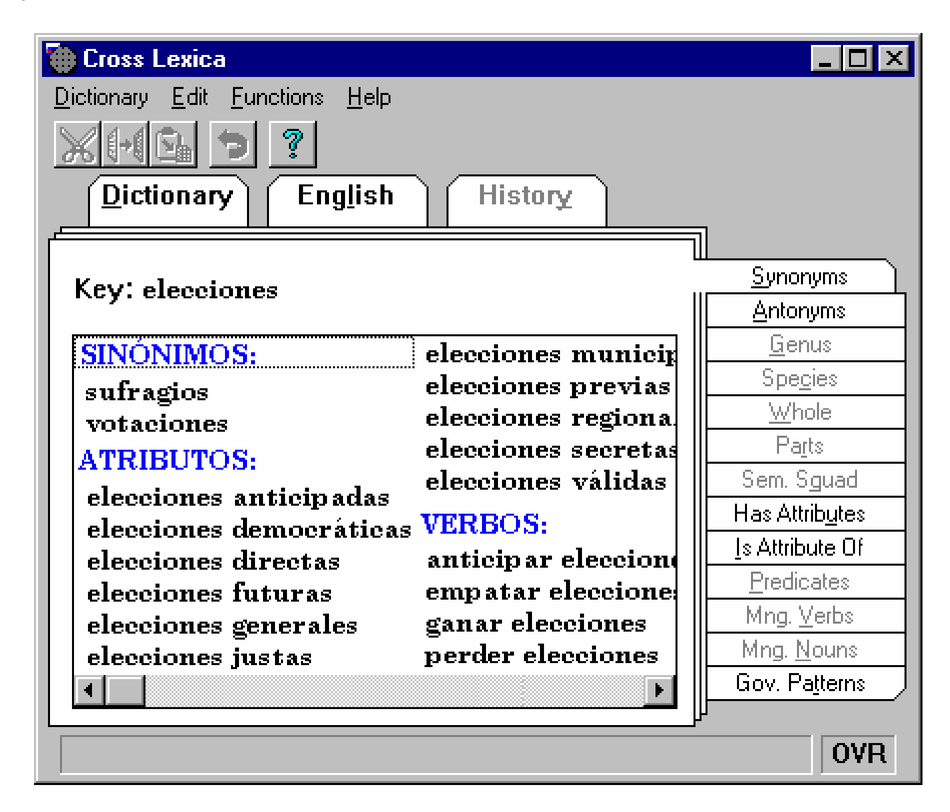


FIGURE III.2. *CrossLexica™, a dictionary of word combinations.*

common combinations is a very important part of linguistic competence of any author. For example, a foreigner might want to know all the verbs commonly used with the Spanish noun *ayuda*, such as *prestar* or *pedir*, or with the noun *atención*, such as *dedicar* or *prestar*, in order to avoid combinations like *pagar atención*, which is a word-by-word translation of the English combination *to pay attention*. Special language-dependent dictionaries are necessary for this purpose (see, for example, Figure III.2).

Within such systems, various complex operations are needed, such as automated reduction of the entered words to their dictionary forms, search of relevant words in the corresponding linguistic database, and displaying all of them in a form convenient to a non-

linguist user. These operations are versatile and include both morphologic and syntactic issues [37].

Another example of a dictionary that provides a number of semantic relations between different lexemes is EuroWordNet [55], a huge lexical resource reflecting diverse semantic links between lexemes of several European languages.

The ideological basis of EuroWordNet is the English dictionary WordNet [41]. English nouns, verbs, adjectives, and adverbs were divided into synonymy groups, or synsets. Several semantic relations were established between synsets: *antonymy* (reference to the "opposite" meaning), *hyponymy* (references to the subclasses), *hyperonymy* (reference to the superclass), *meronymy* (references to the parts), *holonymy* (reference to the whole), etc. Semantic links were established also between synsets of different parts of speech.

The classification hierarchy for nouns is especially well developed within WordNet. The number of hierarchical levels is in average 6 to 7, sometimes reaching 15. The upper levels of the hierarchy form the *ontology*, i.e., a presupposed scheme of human knowledge.

In essence, EuroWordNet is a transportation of the WordNet hierarchy to several other European languages, in particular to Spanish. The upper levels of ontology were obtained by direct translation from English, while for the other levels, additional lexicographic research turned out to be necessary. In this way, not only links between synsets within any involved language were determined, but also links between synsets of a number of different languages.

The efforts invested to the WordNet and EuroWordNet were tremendous. Approximately 25´000 words were elaborated in several languages.

INFORMATION RETRIEVAL

Information retrieval systems (IRS) are designed to search for relevant information in large documentary databases. This information can be of various kinds, with the queries ranging from "Find all the documents containing the word *conjugar*" to "Find information on

the conjugation of Spanish verbs". Accordingly, various systems use different methods of search.

The earliest IRSs were developed to search for scientific articles on a specific topic. Usually, the scientists supply their papers with a set of keywords, i.e., the terms they consider most important and relevant for the topic of the paper. For example, *español, verbos, subjuntivo* might be the keyword set of the article "On means of expressing unreal conditions" in a Spanish scientific journal.

These sets of keywords are attached to the document in the bibliographic database of the IRS, being physically kept together with the corresponding documents or separately from them. In the simplest case, the query should explicitly contain one or more of such keywords as the condition on what the article can be found and retrieved from the database. Here is an example of a query: "Find the documents on *verbos* **and** *español*". In a more elaborate system, a query can be a longer logical expression with the operators **and**, **or**, **not**, e.g.: "Find the documents on (*sustantivos* **or** *adjetivos*) **and** (**not** *inglés*)".

Nowadays, a simple but powerful approach to the format of the query is becoming popular in IRSs for non-professional users: the query is still a set of words; the system first tries to find the documents containing all of these words, then all but one, etc., and finally those containing only one of the words. Thus, the set of keywords is considered in a step-by-step transition from conjunction to disjunction of their occurrences. The results are ordered by degree of *relevance*, which can be measured by the number of relevant keywords found in the document. The documents containing more keywords are presented to the user first.

In some systems the user can manually set a threshold for the number of the keywords present in the documents, i.e., to search for "at least m of n" keywords. With $m = n,$ often too few documents, if any, are retrieved and many relevant documents are not found; with $m = 1$, too many unrelated ones are retrieved because of a high rate of false alarms.

Usually, *recall* and *precision* are considered the main characteristics of IRSs. *Recall* is the ratio of the *number of relevant documents found* divided by the *total number of relevant documents* in the database. *Precision* is the ratio of the *number of relevant documents* divided by the *total number of documents found*.

It is easy to see that these characteristics are contradictory in the general case, i.e. the greater one of them the lesser another, so that it is necessary to keep a proper balance between them.

In a specialized IRS, there usually exists an automated indexing subsystem, which works before the searches are executed. Given a set of keywords, it adds, using the **or** operator, other related keywords, based on a hierarchical system of the scientific, technical or business terms. This kind of hierarchical systems is usually called *thesaurus* in the literature on IRSs and it can be an integral part of the IRS. For instance, given the query "Find the documents on *conjugación*," such a system could add the word *morfología* to both the query and the set of keywords in the example above, and hence find the requested article in this way.

Thus, a sufficiently sophisticated IRS first enriches the sets of keywords given in the query, and then compares this set with the previously enriched sets of keywords attached to each document in the database. Such comparison is performed according to any criteria mentioned above. After the enrichment, the average recall of the IRS system is usually increased.

Recently, systems have been created that can automatically build sets of keywords given just the full text of the document. Such systems do not require the authors of the documents to specifically provide the keywords. Some of the modern Internet *search engines* are essentially based on this idea.

Three decades ago, the problem of automatic extraction of keywords was called *automatic abstracting*. The problem is not simple, even when it is solved by purely statistical methods. Indeed, the most frequent words in any business, scientific or technical texts are purely auxiliary, like prepositions or auxiliary verbs. They do not reflect the essence of the text and are not usually taken for abstract-

ing. However, the border between auxiliary and meaningful words cannot be strictly defined. Moreover, there exist many term-forming words like *system, device,* etc., which can seldom be used for information retrieval because their meaning is too general. Therefore, they are not useful for abstracts.

The multiplicity of IRSs is considered now as an important class of the applied software and, specifically, of applied linguistic systems. The period when they used only individual words as keys has passed. Developers now try to use word combinations and phrases, as well as more complicated strategies of search. The limiting factors for the more sophisticated techniques turned out to be the same as those for grammar and style checkers: the absence of complete grammatical and semantic analysis of the text of documents. The methods used now even in the most sophisticated Internet search engines are not efficient for accurate information retrieval. This leads to a high level of *information noise*, i.e., delivering of irrelevant documents, as well as to the frequent missing of relevant ones.

The results of retrieval operations directly depend on the quality and performance of the indexing and comparing subsystems, on the content of the terminological system or the thesaurus, and other data and knowledge used by the system. Obviously, the main tools and data sets used by an IRS have the linguistic nature.

TOPICAL SUMMARIZATION

In many cases, it is necessary to automatically determine what a given document is about. This information is used to classify the documents by their main topics, to deliver by Internet the documents on a specific subject to the users, to automatically index the documents in an IRS, to quickly orient people in a large set of documents, and for other purposes.

Such a task can be viewed as a special kind of summarization: to convey the contents of the document in a shorter form. While in "normal" summarization by the contents the main ideas of the

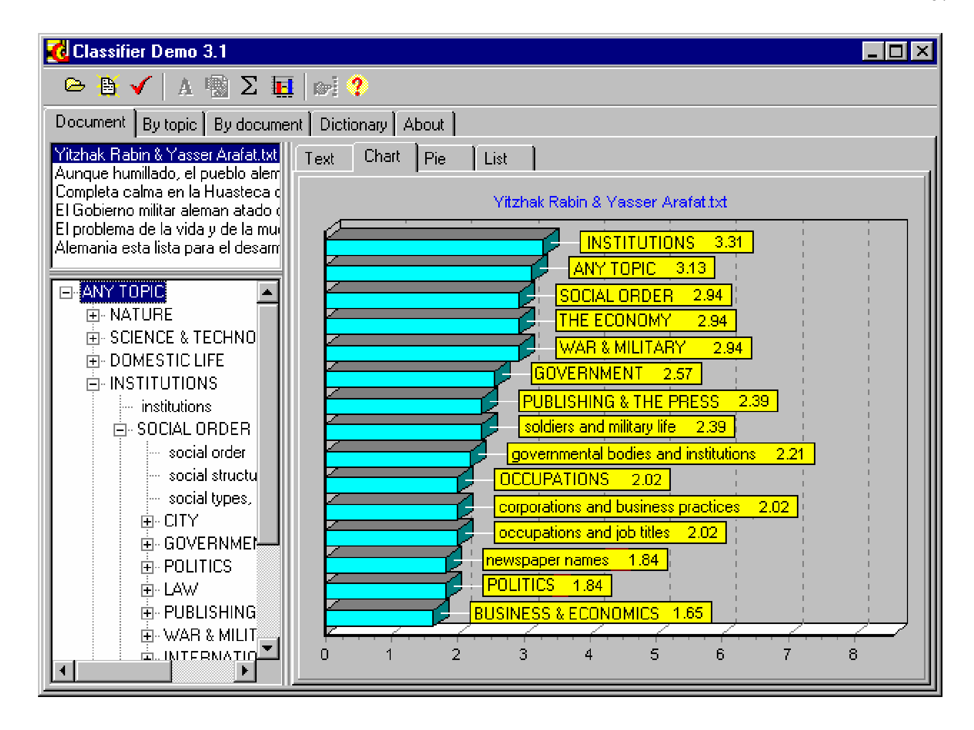


FIGURE III.3. Classifier *program determines the main topics of a document.*

document are considered, here we consider only the topics mentioned in the document, hence the term *topical* summarization.

As an example, let us consider the system Clasitex™ that automatically determines the main topics of a document. A variant of its implementation, Classifier™, was developed in the Center of Computing Research, National Polytechnic Institute at Mexico City [46] (see Figure III.3). It uses two kinds of linguistic information:

- First, it neutralizes morphologic variations in order to reduce any word found in the text to its standard (i.e., dictionary) form, e.g., *oraciones* → *oración*, *regímenes* → *régimen*, *lingüísticas* → *lingüístico*, *propuesto* → *proponer*.
- Second, it puts into action a large dictionary of thesaurus type, which gives, for each word in its standard form, its correspond-

ing position in a pre-defined hierarchy of topics. For example, the word *oración* belongs to the topic *lingüística*, which belongs in turn to the topic *ciencias sociales*, which in its turn belongs to the topic *ciencia*.

Then the program counts how many times each one of these topics occurred in the document. Roughly speaking, the topic mentioned most frequently is considered the main topic of the document. Actually, the topics in the dictionary have different weights of importance [43, 45], so that the main topic is the one with the greatest total weight in the document.

Applied linguistics can improve this method in many possible ways. For example, in its current version, Clasitex does not count any pronouns found in the text, since it is not obvious what object a personal pronoun such as *él* can refer to.

What is more, many Spanish sentences contain zero subjects, i.e. implicit references to some nouns. This becomes obvious in English translation: *Hay un libro. Es muy interesante* ⇒ *There is a book.* ***It*** *is very interesting* ⇒ ***El libro*** *es muy interesante.* Thus, each Spanish sentence without any subject is implicitly an occurrence of the corresponding word, which is not taken into account by Clasitex, so that the gathered statictics is not completely correct.

Another system, TextAnalyst™, for determining the main topics of the document and the relationships between words in the document was developed by MicroSystems, in Russia (see Figure III.4). This system is not dictionary-based, though it does have a small dictionary of stop-words (these are prepositions, articles, etc., and they should not be processed as meaningful words).

This system reveals the relationships between words. Words are considered related to each other if they co-occurred closely enough in the text, e.g., in the same sentence. The program builds a network of the relationships between words. Figure III.4 shows the most important words found by TextAnalyst in the early draft of this book, and the network of their relationships.

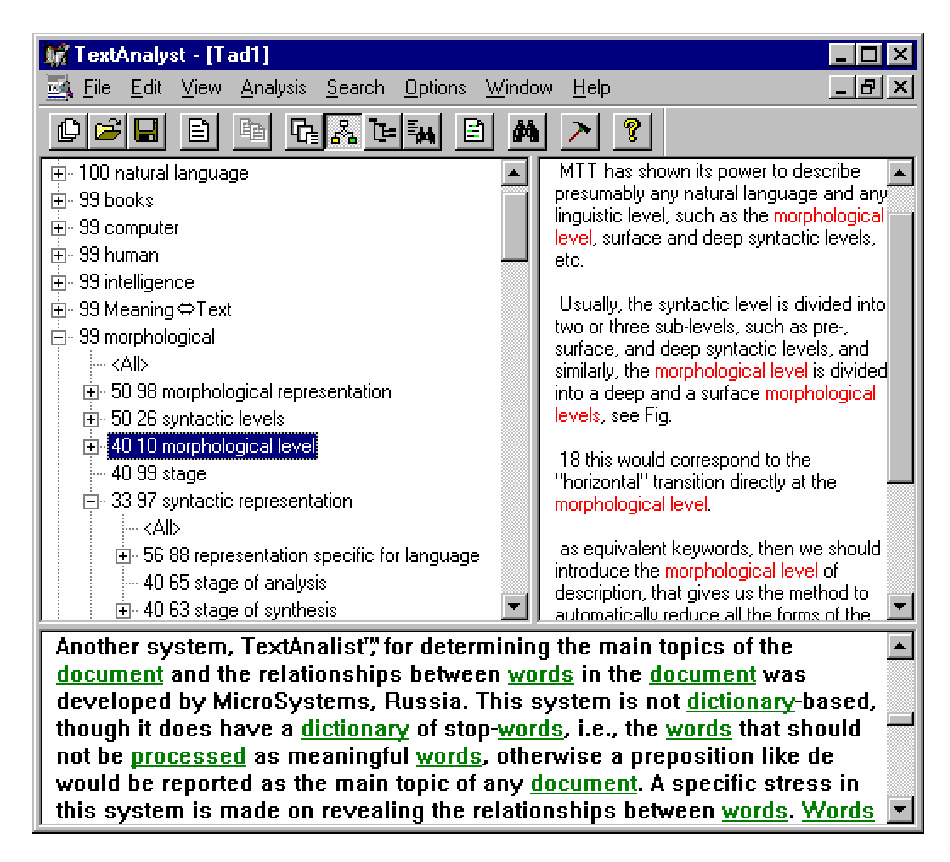


FIGURE III.4. TextAnalyst *program reveals the relationships between words.*

As in Clasitex, the degree of importance of a word, or its weight, is determined in terms of its frequency, and the relationships between words are used to mutually increase the weights. The words closely related to many of the important words in the text are also considered important.

In TextAnalyst, the list of the important words is used for the following tasks:

- *Compression of text* by eliminating the sentences or paragraphs that contain the minimal number of important words, until the size of the text reaches the threshold selected by the user,

- *Building hypertext* by constructing mutual references between the most important words and from the important words to others to which they are supposedly related.

The TextAnalyst technology is based on a special type of a dynamic neural network algorithm. Since the Clasitex program is based on a large dictionary, it is a knowledge-based program, whereas TextAnalyst is not.

AUTOMATIC TRANSLATION

Translation from one natural language to another is a very important task. The amount of business and scientific texts in the world is growing rapidly, and many countries are very productive in scientific and business domains, publishing numerous books and articles in their own languages. With the growth of international contacts and collaboration, the need for translation of legal contracts, technical documentation, instructions, advertisements, and other texts used in the everyday life of millions of people has become a matter of vital importance.

The first programs for automatic, or machine, translation were developed more than 40 years ago. At first, there existed a hope that texts could be translated word by word, so that the only problem would be to create a dictionary of pairs of words: a word in one language and its equivalent in the other. However, that hope died just after the very first experiments.

Then the ambitious goal was formulated to create programs which could understand deeply the meaning of an arbitrary text in the source language, record it in some universal intermediate language, and then reformulate this meaning in the target language with the greatest possible accuracy. It was supposed that neither manual pre-editing of the source text nor manual post-editing of the target text would be necessary. This goal proved to be tremendously difficult to achieve, and has still not been satisfactorily accomplished in any but the narrowest special cases.

At present there is a lot of translation software, ranging from very large international projects being developed by several institutes or even several corporations in close cooperation, to simple automatic dictionaries, and from laboratory experiments to commercial products. However, the quality of the translations, even for large systems developed by the best scientists, is usually conspicuously lower than the quality of manual human translation.

As for commercial translation software, the quality of translation it generates is still rather low. A commercial translator can be used to allow people quite unfamiliar with the original language of the document to understand its main idea. Such programs can help in manual translation of texts. However, post-editing of the results, to bring them to the degree of quality sufficient for publication, often takes more time than just manual translation made by a person who knows both languages well enough.[4] Commercial translators are quite good for the texts of very specific, narrow genres, such as weather reports. They are also acceptable for translation of legal contracts, at least for their formal parts, but the paragraphs specifying the very subject of the contract may be somewhat distorted.

To give the reader an impression of what kind of errors a translation program can make, it is enough to mention a well-known example of the mistranslation performed by one of the earliest systems in 1960s. It translated the text from Bible *The spirit is willing, but the flesh is weak* (Matt. 26:41) into Russian and then back into English. The English sentence then turned out to be *The vodka is strong, but the meat is rotten* [34]. Even today, audiences at lectures on automatic translation are entertained by similar examples from modern translation systems.

Two other examples are from our own experience with the popular commercial translation package PowerTranslator by Globalink, one of the best in the market. The header of an English document *Plans* is translated into Spanish as the verb *Planifica*, while the cor-

[4] According to the tests conducted by the publishing house of the Russian edition of PC World.

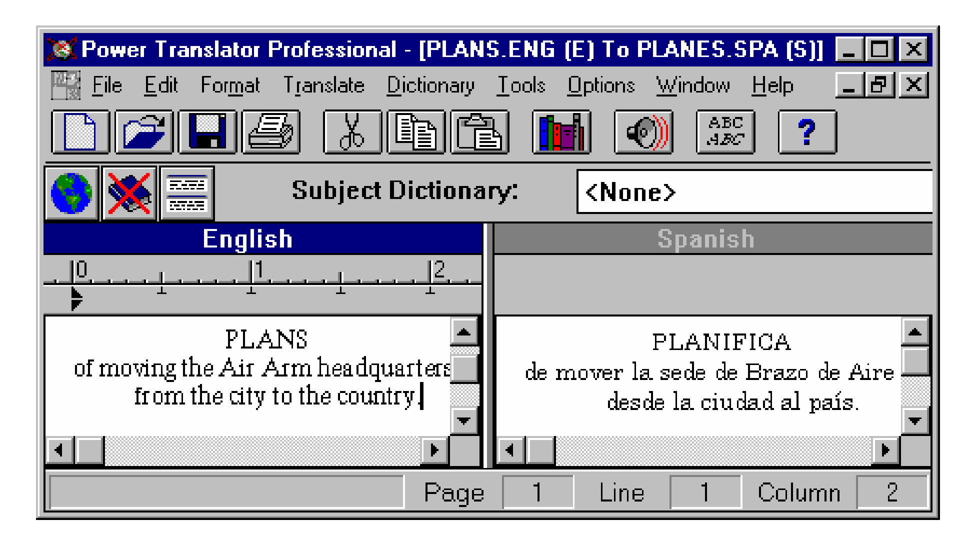


FIGURE III.5. *One of commercial translators.*

rect translation is the Spanish noun *Planes* (see Figure III.5). The Spanish phrase *el papel de Francia en la guerra* is translated as *the paper of France in the war,* while the correct translation is *the role of France in the war*. There are thousands of such examples, so that nearly any automatically translated document is full of them and should be reedited.

Actually, the quality of translation made by a given program is not the same in the two directions, say, from English to Spanish and from Spanish to English. Since automatic analysis of the text is usually a more difficult task than generation of text, the translation *from* a language that is studied and described better has generally higher quality than translation *into* this language. Thus, the elaboration of Spanish grammars and dictionaries can improve the quality of the translation from Spanish into English.

One difficult problem in automatic translation is the word sense disambiguation. In any bilingual dictionary, for many source words, dozens of words in the target language are listed as translations, e.g., for simple Spanish word *gato*: cat, moneybag, jack, sneak thief, trigger, outdoor market, hot-water bottle, blunder, etc. Which one should the program choose in any specific case? This problem has

proven to be extremely difficult to solve. Deep linguistic analysis of the given text is necessary to make the correct choice, on the base on the meaning of the surrounding words, the text, as a whole, and perhaps some extralinguistic information [42].

Another, often more difficult problem of automatic translation is restoring the information that is contained in the source text implicitly, but which must be expressed explicitly in the target text. For example, given the Spanish text *José le dio a María un libro*. *Es interesante*, which translation of the second sentence is correct: *He is interesting*, or *She is interesting*, or *It is interesting*, or *This is interesting*? Given the English phrase *computer shop*, which Spanish translation is correct: *tienda de computadora* or *tienda de computadoras*? Compare this with *computer memory.* Is *they are beautiful* translated as *son hermosos* or *son hermosas*? Is *as you wish* translated as *como quiere, como quieres*, *como quieren*, or *como queréis*?[5] Again, deep linguistic analysis and knowledge, rather than simple word-by-word translation, is necessary to solve such problems.

Great effort is devoted in the world to improve the quality of translation. As an example of successful research, the results of the Translation group of Information Science Institute at University of South California can be mentioned [53]. This research is based on the use of statistical techniques for lexical ambiguity resolution.

Another successful team working on automatic translation is that headed by Yu. Apresian in Russia [34]. Their research is conducted in the framework of the Meaning ⇔ Text model.

NATURAL LANGUAGE INTERFACE

The task performed by a natural language interface to a database is to understand questions entered by a user in natural language and to provide answers—usually in natural language, but sometimes as a

[5] In the variant of Spanish spoken in Spain. Consider also *como vosotros queréis* vs. *como vosotras queréis*.

formatted output. Typically, the entered queries, or questions, concern some facts about data contained in a database.

Since each database is to some degree specialized, the language of the queries and the set of words used in them are usually very limited. Hence, the linguistic task of grammatical and semantic analysis is much simpler than for other tasks related to natural language, such as translation.

There are some quite successful systems with natural language interfaces that are able to understand a very specialized sublanguage quite well. Other systems, with other, usually less specialized sublanguages, are much less successful. Therefore, this problem does not have, at least thus far, a universal solution, most of the solutions being constructed *ad hoc* for each specific system.

The developers of the most popular database management systems usually supply their product with a formal query-constructing language, such as SQL. To learn such a language is not too difficult, and this diminishes the need for a natural language interface. We are not aware of any existing commercial interface system that works with a truly unlimited natural language.

Nevertheless, the task of creating such an interface seems very attractive for many research teams all over the world. Especially useful could be natural language interfaces with speech recognition capabilities, which also would allow the user to make queries or give commands over a telephone line.

The task of development of natural language interfaces, though being less demanding to such branches of linguistics as morphology or syntax, are very demanding to such "deeper" branches of linguistics as semantics, pragmatics, and theory of discourse.

The specific problem of the interface systems is that they work not with a narrative, a monologue, but with a dialogue, a set of short, incomplete, interleaving remarks. For example, in the following dialogue:

User: Are there wide high-resolution matrix printers in the store?
System: No, there are no such printers in the store.
User: And narrow?

it is difficult for the computer to understand the meaning of the last remark.

A rather detailed linguistic analysis is necessary to re-formulate this user's question to *Are there narrow high-resolution matrix printers in the store?* In many cases, the only way for the computer to understand such *elliptical* questions is to build a model of the user's current goals, its knowledge, and interests, and then try to guess what the computer itself would be asking at this point of the dialogue if it were the user, and in what words it would formulate such a question. This idea can be called *analysis through synthesis*.

EXTRACTION OF FACTUAL DATA FROM TEXTS

Extraction of factual data from texts is the task of automatic generation of elements of a factographic database, such as fields, or parameters, based on on-line texts. Often the flows of the current news from the Internet or from an information agency are used as the source of information for such systems, and the parameters of interest can be the demand for a specific type of a product in various regions, the prices of specific types of products, events involving a particular person or company, opinions about a specific issue or a political party, etc.

The decision-making officials in business and politics are usually too busy to read and comprehend all the relevant news in their available time, so that they often have to hire many news summarizers and readers or even to address to a special information agency. This is very expensive, and even in this case the important relationships between the facts may be lost, since each news summarizer typically has very limited knowledge of the subject matter. A fully effective automatic system could not only extract the relevant facts much faster, but also combine them, classify them, and investigate their interrelationships.

There are several laboratory systems of that type for business applications, e.g., a system that helps to explore news on Dow Jones index, investments, and company merge and acquisition projects.

Due to the great difficulties of this task, only very large commercial corporations can afford nowadays the research on the factual data extraction problem, or merely buy the results of such research.

This kind of problem is also interesting from the scientific and technical point of view. It remains very topical, and its solution is still to be found in the future. We are not aware of any such research in the world targeted to the Spanish language so far.

TEXT GENERATION

The generation of texts from pictures and formal specifications is a comparatively new field; it arose about ten years ago. Some useful applications of this task have been found in recent years. Among them are multimedia systems that require a text-generating subsystem to illustrate the pictures through textual explanations. These subsystems produce coherent texts, starting from the features of the pictures.

Another very important application of systems of this kind is the generation of formal specifications in text form from quite formal technical drawings.

For example, compilation of a patent formula for a new device, often many pages long, is a boring, time-consuming, and error-prone task for a human. This task is much more suitable for a machine.

A specific type of such a system is a multilingual text generating system. In many cases, it is necessary to generate descriptions and instructions for a new device in several languages, or in as many languages as possible.

Due to the problems discussed in the section on translation, the quality of automatic translation of a manually compiled text is often very low.

Better results can be achieved by automatic generation of the required text in each language independently, from the technical

drawings and specifications or from a text in a specific formal language similar to a programming language.

Text generating systems have, in general, half of the linguistic problems of a translation system, including all of the linguistic problems connected with the grammar and lexicon of the target language. This is still a vast set of linguistic information, which is currently available in adequate detail for only a few very narrow subject areas.

SYSTEMS OF LANGUAGE UNDERSTANDING

Natural language understanding systems are the most general and complex systems involving natural language processing. Such systems are universal in the sense that they can perform nearly all the tasks of other language-related systems, such as grammar and style checking, information retrieval, automatic translation, natural language interface, extraction of factual data from texts, text generation, and so forth.

For example, automatic translation can be implemented as a text understanding system, which understands the text in the source language and then generates a precise description of the learned information in the target language.

Hence, creation of a text understanding system is the most challenging task for the joint efforts of computational linguistics and artificial intelligence.

To be more precise, the natural language processing module is only one part of such a system. Most activities related to logical reasoning and understanding proper are concentrated in another its part—a reasoning module. These two modules, however, are closely interrelated and they should work in tight cooperation.

The linguistic subsystem is usually bi-directional, i.e., it can both "understand," or *analyze*, the input texts or queries, and produce, or *generate*, another text as the answer. In other words, this subsystem transforms a human utterance into an internal, or semantic, representation comprehensible to the reasoning subsystem, produces a

response in its own internal format, and then transforms the response into a textual answer.

In different systems, the reasoning subsystem can be called the knowledge-based reasoning engine, the problem solver, the expert system, or something else, depending on the specific functions of the whole system. Its role in the whole system of natural language understanding is always very important.

Half a century ago, Alan Turing suggested that the principal test of intelligence for a computer would be its ability to conduct an intelligent dialogue, making reasonable solutions, giving advice, or just presenting the relevant information during the conversation.

This great goal has not been achieved thus far, but research in this direction has been conducted over the past 30 years by specialists in artificial intelligence and computational linguistics.

In order to repeat, the full power of linguistic science and the full range of linguistic data and knowledge are necessary to develop what we can call a true language understanding system.

RELATED SYSTEMS

There are other types of applications that are not usually considered systems of computational linguistics proper, but rely heavily on linguistic methods to accomplish their tasks. Of these we will mention here two, both related to pattern recognition.

Optical character recognition systems recognize the *graphemes*, i.e., letters, numbers, and punctuation marks, in a point-by-point image of an arbitrary text printed on paper, and convert them to the corresponding ASCII codes. The graphemes can be in any font, typeface or size; the background of the paper can contain some dots and spots. An example of what the computer sees is given in Figure III.6.

A human being easily reads the following text *Reglas de interpretación semántica*: **Reglas de interpretación semántica** because he or she understands the meaning of the words. However, without understanding the meaning it is not possible to recognize, say, the first

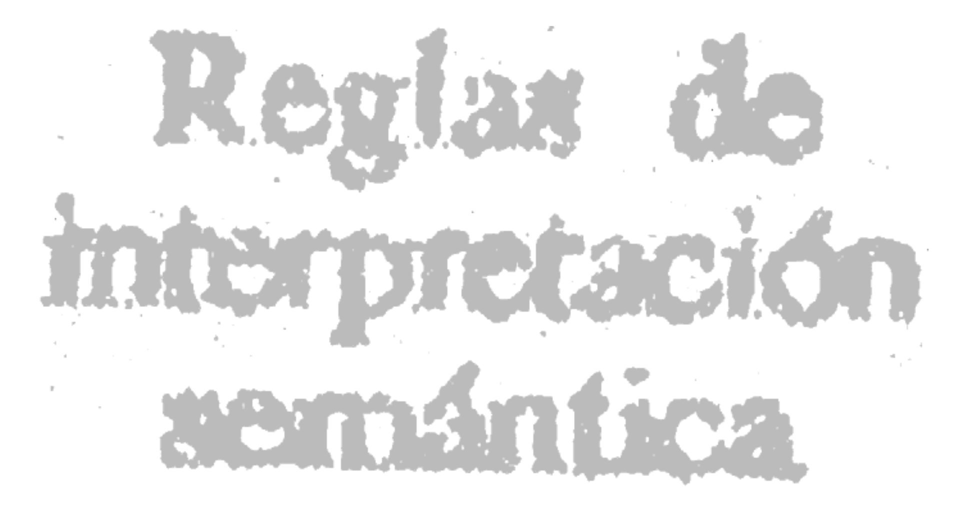


FIGURE III.6. *The image of a text, as the computer sees it.*

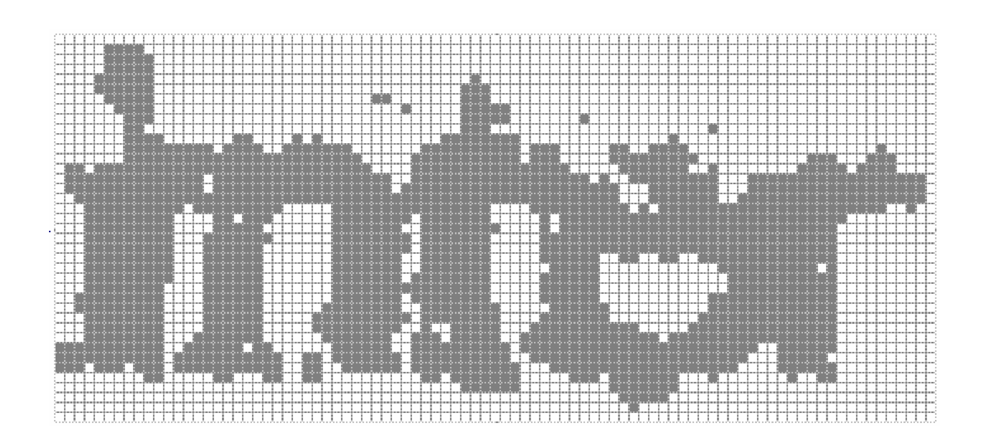


FIGURE III.7. *Several letters of the same text, as the computer sees them.*

letter of the last string (is it **a, s** or **g**?), or the first letter(s) of the second line (is it **r, i** or **m**?).

The first letters of the second line are shown separately in Figure III.7. One who does not know what the whole word means, cannot even say for sure that this picture represents any letters. However, one can easily read precisely the same image, shown

above, of the same letters in their complete context. Hence, it is obvious that the task of optical character recognition cannot be solved only by the methods of image recognition, without linguistic information.

The image recognition proper is beyond the competence of computational linguistics. However, after the recognition of an image even of much higher quality than the one shown in Figure III.6, some peculiar errors can still appear in the textual representation of the image. They can be fixed by the operations similar to those of a spell checker. Such a specialized spell checker should know the most frequent errors of recognition. For example, the lower-case letter *l* is very similar to the figure *1*, the letter *n* is frequently recognized as the pair *ii*, while the *m* can be recognized as *iii* or *rn.* Vice versa, the digraphs *in*, *rn* and *ni* are frequently recognized as *m*, and so forth.

Most such errors can be corrected without human intervention, on the base of linguistic knowledge. In the simplest case, such knowledge is just a dictionary of words existing in the language. However, in some cases a deeper linguistic analysis is necessary for disambiguation. For example, only full parsing of the context sentence can allow the program to decide whether the picture recognized as **danios* actually represents the existing Spanish words *darnos* or *damos*.

A much more challenging task than recognition of printed texts is *handwriting recognition*. It is translation into ASCII form of the texts written by hand with a pen on paper or on the surface of a special computer device, or directly with a mouse on the computer screen. However, the main types of problem and the methods of solution for this task are nearly the same as for printed texts, at least in their linguistic aspect.

Speech recognition is another type of recognition task employing linguistic methods. A speech recognition system recognizes specific sounds in the flow of a speech of a human and then converts them into ASCII codes of the corresponding letters. The task of recognition itself belongs both to pattern recognition and to *phonology*, the

science bordering on linguistics, acoustics, and physiology, which investigates the sounds used in speech.

The difficulties in the task of speech recognition are very similar or quite the same as in optical character recognition: mutilated patterns, fused patterns, disjoint parts of a pattern, lost parts of the pattern, noise superimposing the pattern. This leads to even a much larger number of incorrectly recognized letters than with optical character recognition, and application of linguistic methods, generally in the same manner, is even more important for this task.

CONCLUSIONS

A short review of applied linguistic systems has shown that only very simple tasks like hyphenation or simple spell checking can be solved on a modest linguistic basis. All the other systems should employ relatively deep linguistic knowledge: dictionaries, morphologic and syntactic analyzers, and in some cases deep semantic knowledge and reasoning. What is more, nearly all of the discussed tasks, even spell checking, have to employ very deep analysis to be solved with an accuracy approaching 100%. It was also shown that most of the language processing tasks could be considered as special cases of the general task of language understanding, one of the ultimate goals of computational linguistics and artificial intelligence.

IV. LANGUAGE AS A MEANING ⇔ TEXT TRANSFORMER

IN THIS CHAPTER, we will return to linguistics, to make a review of several viewpoints on natural language, and to select one of them as the base for further studies. The components of the selected approach will be defined, i.e., *text* and *meaning*. Then some conceptual properties of the linguistic transformations will be described and an example of these transformations will be given.

POSSIBLE POINTS OF VIEW ON NATURAL LANGUAGE

One could try to define natural language in one of the following ways:

- The principal means for expressing human thoughts;
- The principal means for text generation;
- The principal means of human communication.

The first definition—"the principal means for expressing human thoughts"—touches upon the expressive function of language. Indeed, some features of the outer world are reflected in the human brain and are evidently processed by it, and this processing is just the human thought. However, we do not have any real evidence that human beings directly use words of a specific natural language in the process of thinking. Modes of thinking other than linguistic ones are also known. For example, mathematicians with different native languages can have the same ideas about an abstract subject, though they express these thoughts in quite different words. In addition, there are kinds of human thoughts—like operations with musical or visual images—that cannot be directly reduced to words.

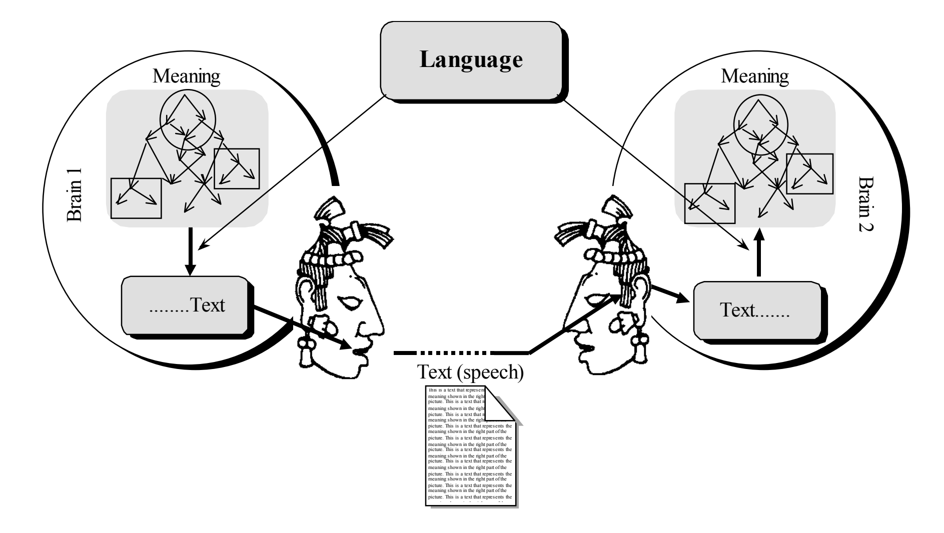


FIGURE IV.1. *The role of language in human communication.*

As to the second definition—"the principal means for text generation"—there is no doubt that the flow of utterances, or texts, is a very important result of functioning of natural language.

However, communication includes not only generation (speaking), but also understanding of utterances. This definition also ignores the starting point of text generation, which is probably the target point of understanding. Generation cannot exist without this starting point, which is contents of the target utterance or the text in the whole. We can call these contents *meaning*.

As to the third definition—"the principal means of human communication,"—we can readily agree that the communicative function is the main function of natural language.

Clearly, only persons living in contact with society really need a language for efficient communication. This definition is perhaps correct, but it does not touch upon two main aspects of communication in natural language, namely, speaking and understanding, and thus it does not try to define these aspects, separately and in their interaction.

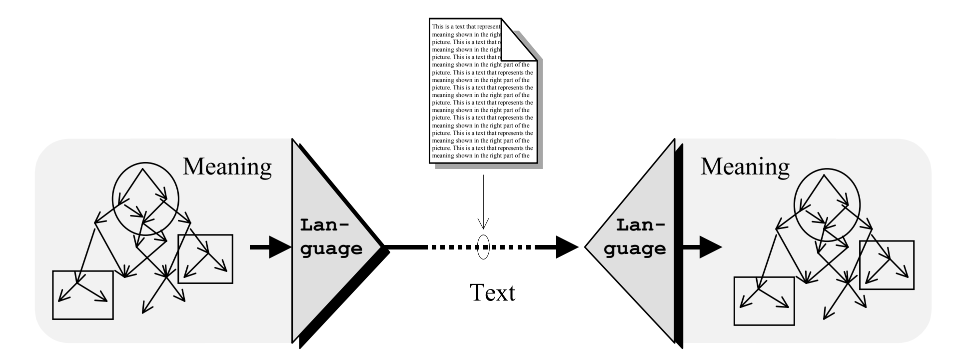


FIGURE IV.2. *Language functions like encoder / decoder in a communication channel.*

Thus, these definitions are not sufficient for our purposes. A better definition should touch upon all useful components of the ones given above, such as *text*, *meaning*, *generation*, and *understanding*. Such definition will be given in the next section.

LANGUAGE AS A BI-DIRECTIONAL TRANSFORMER

The main purpose of human communication is transferring some information—let us call it Meaning[6]—from one person to the other. However, the direct transferring of thoughts is not possible.

Thus, people have to use some special physical representation of their thoughts, let us call it Text.[7] Then, language is a tool to transform one of these representations to another, i.e. to transform Meanings to words when speaking, and the words to their Meaning when listening (see Figure IV.1).

[6] We use capitalization to distinguish the terms Meaning and Text in their specific sense used in the Meaning ⇔ Text Theory from the conventional usage of these words.

[7] For the sake of the argument, here we consider speech, as well as written text, to be a kind of Text.

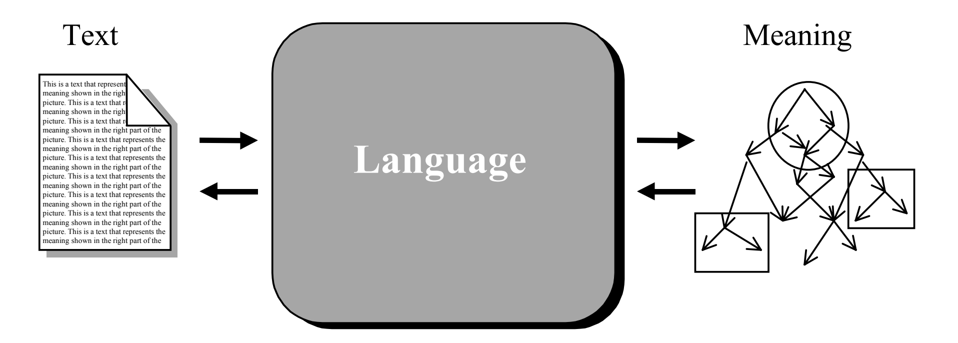


FIGURE IV.3. *Language as a Meaning ⇔ Text transformer.*

It is important to realize that the communicating persons use the same language, which is their common knowledge, and each of them has a copy of it in the brain.

If we compare this situation with transferring the information over a communication channel, such as a computer network, the role of language is encoding the information at the transmitting end and then decoding it at the receiving end.[8] Again, here we deal with two copies of the same encoder/decoder (see Figure IV.2).

Thus, we naturally came to the definition of natural language as a transformer of Meanings to Texts, and, in the opposite direction, from Texts to Meanings (see Figure IV.3).

This transformer is supposed to reside in human brain. By transformation we mean some form of translation, so that both the Text and the corresponding Meaning contain the same information. What we specifically mean by these two concepts, Text and Meaning, will be discussed in detail later.

Being originally expressed in an explicit form by Igor Mel'čuk, this definition is shared nowadays by many other linguists. It permits to recognize how computer programs can simulate, or model, the capacity of the human brain to transform the information from one of these representations into another.

[8] It is not surprising that later in this book, levels of information representation will appear, in direct analogy with the modern computer network protocols.

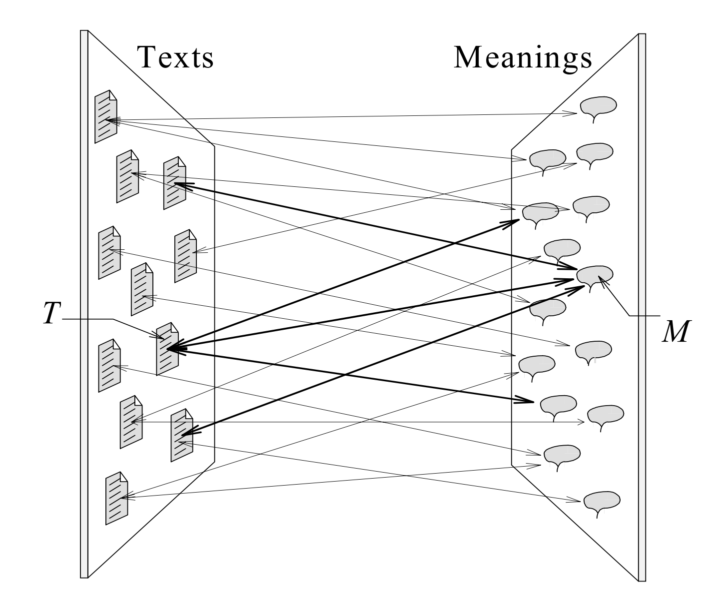


FIGURE IV.4. *Meaning ⇔ Text many-to-many mapping.*

Essentially, this definition combines the second and the third definitions considered in the previous section. Clearly, the transformation of Text into Meaning and vice versa is obligatory for any human communication, since it implies transferring the Meaning from one person to another using the Text as its intermediate representation. The transformation of Meaning into Text is obligatory for the generation of utterances. To be more precise, in the whole process of communication of human thoughts the definition 1 given earlier actually refers to Meaning, the definition 2 to Text, and the definition 3 to both mentioned aspects of language.

With our present definition, language can be considered analogous to a technical device, which has input and output. Some information, namely Text, being entered to its input, is transformed into another form with equivalent contents.

The new form at the output is Meaning. More precisely, we consider a bi-directional transformer, i.e., two transformers working in

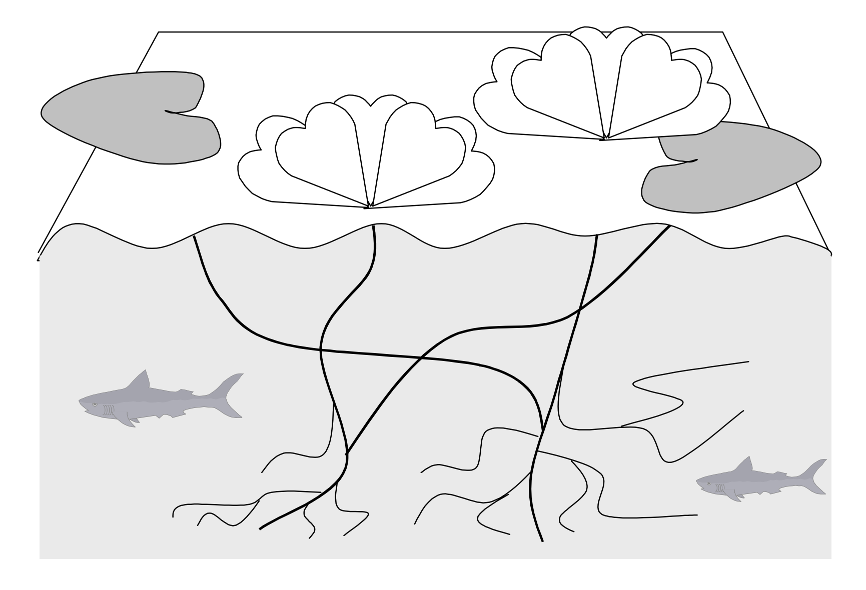


FIGURE IV.5. *Metaphor of surface and deep structures.*

parallel but in opposite directions. Text is the result of the activity of one of these transformers, and Meaning, of the other.

Programmers can compare such a device with a compiler, let us say, a C++ compiler, which takes a character file with the ASCII text of the program in the input and produces some binary code with machine instructions, as the output. The binary code corresponds to the meaning of the program. However, a compiler usually cannot translate the machine instructions back to a C++ program text.

As a mathematical analogy to this definition, we can imagine a bi-directional mapping between one huge set, the set of all possible Texts, and another huge set, the set of all possible Meanings (see Figure IV.4).

The two sets, Texts and Meanings, are not quite symmetric in their properties. Only the Texts have an explicit expression, only they can be immediately observed or directly transferred from one person to another, while the Meanings reside in the brain of each person independently and cannot be immediately observed or assessed.

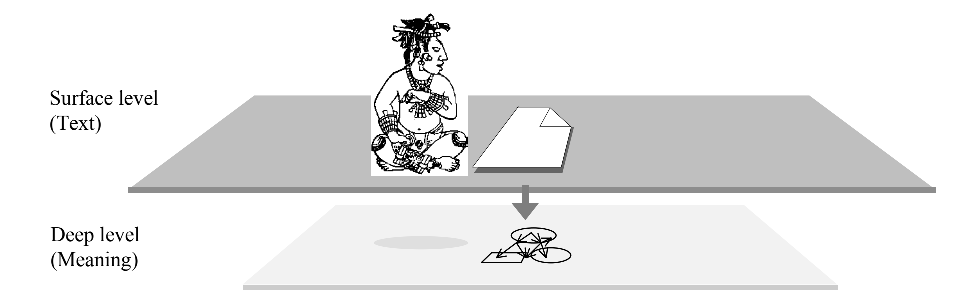


FIGURE IV.6. *Two levels of representation.*

This is similar to the front panel of an electrical device: the lights and switches on its *surface* can be observed, while the electrical processes represented by the lights and controlled by the switches are *deep*[9] under the cover of the device, and can only be guessed by watching the lights or experimenting with the switches.

Another metaphor of surface and deep structures of language is shown in Figure IV.5. We can directly observe the surface of water, but in order to learn what leaf is connected to what flower through common roots, we need to analyze what is under the surface. There is much more below the surface than on top of it, and only analysis of the deeper phenomena gives us understanding of the whole thing.

All this is often considered as surface and deep *levels of the representation* of utterances (see Figure IV.6). The man on the picture cannot see the meaning of the text immediately and has to penetrate below the surface of the text to find its meaning.

Thus, the set of Texts is considered the *surface* edge of the Meaning ⇔ Text transformer, while the set of Meanings gives its *deep* edge. The Meaning corresponding to the given Text at the depth is also called its *semantic representation*.

The transformation of Meaning into Text is called *synthesis* of the Text. The transformation to the inverse direction, that is from Text

[9] See also discussion of the terms *surface* and *deep* in comparison with their usage in generative grammars on the page 124.

into Meaning, is called *analysis* of Text. Thus, according to our basic definition, natural language is both *analyzer* and *synthesizer* of Texts, at the same time.

This definition uses the notions of Text and Meaning, although they have been neither defined nor described so far. Such descriptions will be given in the following sections.

TEXT, WHAT IS IT?

The empirical reality for theoretical linguistics comprises, in the first place, the sounds of speech. Samples of speech, i.e., separate words, utterances, discourses, etc., are given to the researchers directly and, for living languages, are available in an unlimited supply.

Speech is a continuous flow of acoustic signals, just like music or noise. However, linguistics is mainly oriented to the processing of natural language in a discrete form.

The discrete form of speech supposes dividing the flow of the acoustic signals into sequentially arranged entities belonging to a finite set of partial signals. The finite set of all possible partial signals for a given language is similar to a usual alphabet, and is actually called a *phonetic alphabet*.

For representation of the sound of speech on paper, a special *phonetic transcription* using *phonetic symbols* to represent speech sounds was invented by scientists. It is used in dictionaries, to explain the pronunciation of foreign words, and in theoretical linguistics.

A different, much more important issue for modern computational linguistics form of speech representation arose spontaneously in the human practice as the written form of speech, or the writing system.

People use three main writing systems: that of alphabetic type, of syllabic type, and of hieroglyphic type. The majority of humankind use alphabetic writing, which tries to reach correspondence between letters and sounds of speech.

Two major countries, China and Japan,[10] use the hieroglyphic writing. Several countries use syllabic writing, among them Korea. *Hieroglyphs* represent the meaning of words or their parts. At least, they originally were intended to represent directly the meaning, though the direct relationship between a hieroglyph and the meaning of the word in some cases was lost long ago.

Letters are to some degree similar to sounds in their functions. In their origin, letters were intended to directly represent sounds, so that a text written in letters is some kind of representation of the corresponding sounds of speech. Nevertheless, the simple relationship between letters and sounds in many languages was also lost. In Spanish, however, this relationship is much more straightforward than, let us say, in English or French.

Syllabic signs are similar to letters, but each of them represents a whole syllable, i.e., a group of one or several *consonants* and a *vowel*. Thus, such a writing system contains a greater number of signs and sometimes is less flexible in representing new words, especially foreign ones. Indeed, foreign languages can contain specific combinations of sounds, which cannot be represented by the given set of syllables. The syllabic signs usually have more sophisticated shape than in letter type writing, resembling hieroglyphs to some degree.

In more developed writing systems of a similar type, the signs (called in this case *glyphs*) can represent either single sounds or larger parts of words such as syllables, groups of syllables, or entire words. An example of such a writing system is Mayan writing (see Figure I.2). In spite of their unusual appearance, Mayan glyphs are more syllabic signs than hieroglyphs, and they usually represent the sounds of the speech rather than the meaning of words. The reader can become familiar with Mayan glyphs through the Internet site [52].

[10] In fact, Japanese language uses a mixture of hieroglyphic and syllabic symbols, though the use of syllabic symbols is limited.

Currently, most of the practical tasks of computational linguistics are connected with written texts stored on computer media. Among written texts, those written in alphabetic symbols are more usual for computational linguistics than the phonetic transcription of speech.[11] Hence, in this book the methods of language processing will usually be applied to the written form of natural language.

For the given reason, Texts mentioned in the definition of language should then be thought of as common texts in their usual written form. Written texts are chains of letters, usually subdivided into separate words by spaces[12] and punctuation marks. The combinations of words can constitute sentences, paragraphs, and discourses. For computational linguistics, all of them are examples of Texts.[13]

Words are not utmost elementary units of language. Fragments of texts, which are smaller than words and, at the same time, have their own meanings, are called *morphs*. We will define morphs more precisely later. Now it is sufficient for us to understand that a morph can contain an arbitrary number of letters (or now and then no letters at all!), and can cover a whole word or some part of it. Therefore, Meanings can correspond to some specially defined parts of words, whole words, phrases, sentences, paragraphs, and discourses.

It is helpful to compare the linear structure of text with the flow of musical sounds. The mouth as the organ of speech has rather limited abilities. It can utter only one sound at a time, and the flow of these sounds can be additionally modulated only in a very restricted manner, e.g., by stress, intonation, etc. On the contrary, a set of musical instruments can produce several sounds synchronously, form-

[11] This does not mean that the discussed methods are not applicable to phonetic transcription or, on the other hand, to hieroglyphs. However, just for simplification we will choose only the representation by letters.

[12] In some writing systems, like Japanese, words are not separated by spaces in the written text. Of course, this does not mean that these languages do not have words, but the word boundaries are not reflected in writing. As opposed to Japanese, Vietnamese separates all the syllables.

[13] In Western tradition including HPSG, Text in the given sense is called *list of phoneme strings*, or simply phonetic representation of a linguistic sign.

ing harmonies or several melodies going in parallel. This parallelism can be considered as nonlinear structuring. The human had to be satisfied with the instrument of speech given to him by nature. This is why we use while speaking a linear and rather slow method of acoustic coding of the information we want to communicate to somebody else.

The main features of a Text can be summarized as follows:

- *Meaning*. Not any sequence of letters can be considered a text. A text is intended to encode some information relevant for human beings. The existing connection between texts and meanings is the reason for processing natural language texts.
- *Linear structure*. While the information contained in the text can have a very complicated structure, with many relationships between its elements, the text itself has always one-dimensional, linear nature, given letter by letter. Of course, the fact that lines are organized in a square book page does not matter: it is equivalent to just one very long line, wrapped to fit in the pages. Therefore, a text represents non-linear information transformed into a linear form. What is more, the human cannot represent in usual texts even the restricted non-linear elements of spoken language, namely, intonation and logical stress. Punctuation marks only give a feeble approximation to these non-linear elements.
- *Nested structure* and *coherence*. A text consists of elementary pieces having their own, usually rather elementary, meaning. They are organized in larger structures, such as words, which in turn have their own meaning. This meaning is determined by the meaning of each one of their components, though not always in a straightforward way. These structures are organized in even larger structures like sentences, etc. The sentences, paragraphs, etc., constitute what is called *discourse*, the main property of which is its *connectivity*, or *coherence*: it tells some consistent story about objects, persons, or relations, common to all its parts. Such organization provides linguistics with the means to develop the methods of intelligent text processing.

Thus, we could say that linguistics studies *human* ways of *linear* encoding[14] of *non-linear* information.

MEANING, WHAT IS IT?

Meanings, in contrast to texts, cannot be observed directly. As we mentioned above, we consider the Meaning to be the structures in the human brain which people experience as ideas and thoughts. Since we do not know and cannot precisely represent those brain processes, for practical purposes we must use a representation of Meaning, which is more suitable for manipulation in a computer. Thus, for our purposes, Meaning will be identified with that representation.

In the future, neurophysiological research will eventually discover what signals really correspond to meanings in our brain, and what structure these signals have, but for now those signals remain only vaguely understood. Hence we take the pragmatic viewpoint that, if a representation we use allows a computer to manipulate and respond to texts with an ability close to that of a human, then this representation is rather good for the real Meaning and fits our purposes.

As it was described earlier, the task of language is to transform information from one representation, the Text, into another, the Meaning, and vice versa. A computer program that models the function of language must perform the same task. In any application, there should be some other system or device that consumes the results of the transformation of texts, and produces the information that is to be transformed into a text. The operations of such a device or a system is beyond the scope of computational linguistics itself. Rather, such a system *uses* the linguistic module as its interface with the outer world (see Figure IV.7).

[14] And also compression, which increases the non-linearity of the underlying structure.

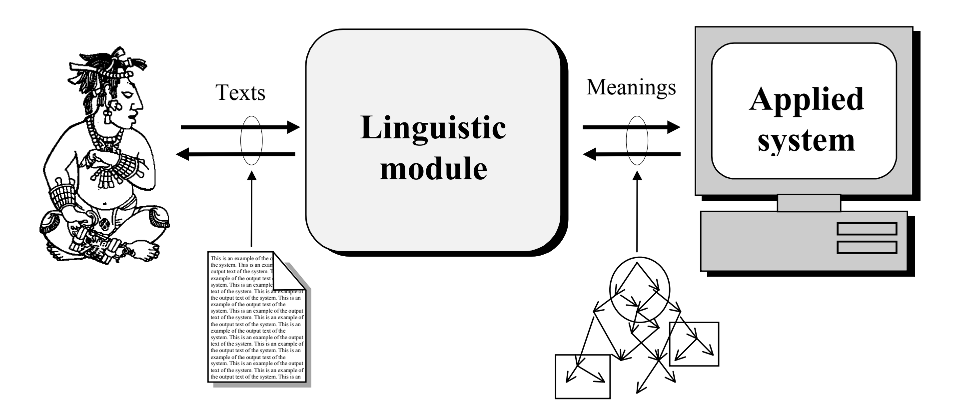


FIGURE IV.7. *Structure of an application system with a natural language interface.*

For a linguistic module of a system, the Meaning is a formal language or a format of information representation immediately understandable for, or executable on, the consumer of the information: the underlying expert or reasoning system, database, robot control system, etc. It is supposed that this underlying system produces its responses just in the same format. Thus, in practice, the format of Meaning is already given to the developers of the linguistic module for any specific application system.

Usually, such systems are aware on the *entities* mentioned in the text, their *states, properties*, *processes*, *actions*, and *relationships*.

Besides, there exists other type of information in a text, such as *beliefs, estimations,* and *intentions* of its author. For example, in the Spanish sentence ***Creo que*** *su esposa está aquí,* the part reflecting the author's belief is given in bold face. The author usually flavors any text with these elements. Words reflecting basic information, through some stylistic coloring, can additionally express author's attitude. For example, the Spanish sentence *Este* ***hombrón*** *no está trabajando ahora* has the meaning 'this man is not working now and I consider him big and coarse'.

Perhaps only very formal texts like legal documents do not contain subjective attitude of the author(s) to the reflected issues. The

advanced application system should distinguish the basic information delivered in texts from author's beliefs, estimations, and intentions.

Additionally, even a very formal text contains many explicit references and explanations of links between parts of a text, and these elements serve as a content table or a protocol of the author's information about text structuring. This is not the information about the relevant matters as such. Instead, this is some kind of meta-information about how the parts of the text are combined together, i.e., a "text about text." We will not discuss such insertions in this book. Since properties, processes, and actions can be represented as relationships between entities touched upon in Text, just these features are used to represent Meaning.

The entities, states, properties, processes, and actions are usually denoted by some names in semantic representation. These names can be compared with the names of variables and functions in a computer program. Such names have no parts of speech, so that a process or an action of, say, 'development' can be equally called in Spanish *desarrollar* or *desarrollo*, while the property of 'small' can be equally called *pequeño* or *ser pequeño*. Usually only one wordform is used for the name, so that the property itself is called, for instance, *pequeño* (neither *pequeña* nor *pequeñas*). Concerning the value *plural* of grammatical category of number, on the semantic level it is transformed to the notion of multiplicity and is represented by a separate element.

TWO WAYS TO REPRESENT MEANING

To represent the entities and relationships mentioned in the texts, the following two logically and mathematically equivalent formalisms are used:

- *Predicative formulas*. *Logical predicates* are introduced in mathematical logic. In linguistics, they are used in conventional logical notation and can have one or more arguments. Coherence

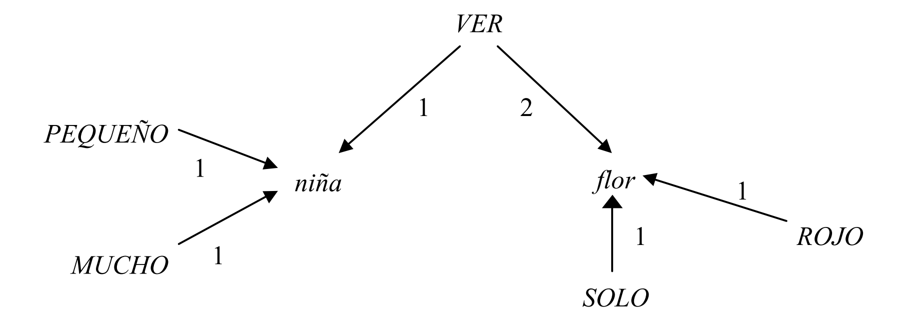


FIGURE IV.8. *Semantic network for the sentence*
Las niñas pequeñas ven la flor roja.

of a text is expressed in the first place by means of common arguments for the predicates involved. For example, the meaning of the sentence *Las niñas pequeñas ven la flor roja* is represented by the following conjunctive predicative formula:

> *VER*(*niña, flor*) &
> *MUCHO*(*niña*) &
> *PEQUEÑO*(*niña*) &
> *SOLO*(*flor*) &
> *ROJO*(*flor*)

In such representation, predicates *SOLO* and *MUCHO* have the meanings 'number of the entities given by the argument is one' and 'number of the entities given by the argument is more than one,' respectively. Arguments of the predicates that are not predicates by themselves are called *terms*. They are written in lowercase letters, in contrast to the predicates that are written in uppercase.

- *Directed labeled graphs.* The nodes of these graphs represent the terms or predicates, and the arrows connect predicates with their arguments, i.e., terms or other predicates. The arrows are marked with numeric labels according to the number of the corresponding argument (1^{st} argument, 2^{nd}, etc.). Though each predicate as-

signs to each its argument a specific *semantic role*, the numerical labels in the graph are used only to distinguish the arguments. For predicates denoting actions, the label 1 usually marks the agent, the label 2, patient or target, etc. Nevertheless, a label of such type does not make any semantic allusion, the enumeration being rather arbitrary. Thus, the graph representation is just equivalent to the predicate one rather than provides any additional information. The semantic representation in the form of directed labeled graph is often called *semantic network*. Figure IV.8 shows the semantic network representation for the example above.

These two representations are equivalent, and either of them can be used. Indeed, there exist a number of easy ways to encode any network linearly.

For human readers, in books and articles, the graph representation is especially convenient. For internal structures of a computer program, the equivalent predicative representation is usually preferred, with special formal means to enumerate the referentially common arguments of various logical predicates.

The graph representation explicitly shows the commonality of the arguments, making it obvious what is known about a specific entity. In the drawing shown in the Figure IV.8, for example, it is immediately seen that there are three pieces of information about the terms *niña* and *flor*, two pieces about the predicate *VER*, and one for predicate *ROJO* and *SOLO*.

Some scientists identify the representations of Meaning with the representation of human knowledge in general[15]. The human knowledge is of concern not only for natural language processing, but also for the task of transferring knowledge between computers, whether that knowledge is expressed by natural language or by some other means. For the transfer of knowledge, it is important to standardize

[15] Within the computer community, efforts are under way to develop knowledge representation both in a linear format (KIF = Knowledge Interchange Format), and in a graphical format (CG = Conceptual Graphs).

methods of representation, so that the knowledge can be communicated accurately. Just for these purposes, the computer community is developing knowledge representation in both the linear and the graphical format. The accuracy and utility of any representation of knowledge should be verified in practice, i.e., in applications.

In the opinion of other scientists, the representations of Meaning and of human knowledge can operate by the same logical structures, but the knowledge in general is in no way coincident with purely linguistic knowledge and even can have different, non-discrete, nature. Hence, they argue, for the transition from Meaning in its linguistic sense to the corresponding representation in terms of general human knowledge, some special stage is needed. This stage is not a part of language and can operate with tools not included in those of the language proper.

DECOMPOSITION AND ATOMIZATION OF MEANING

Semantic representation in many cases turns out to be *universal*, i.e., common to different natural languages. Purely grammatical features of different languages are not usually reflected in this representation. For example, the gender of Spanish nouns and adjectives is not included in their semantic representation, so that this representation turned to be equal to that of English. If the given noun refers to a person of a specific sex, the latter is reflected on semantic level explicitly, via a special predicate of sex, and it is on the grammar of specific language where is established the correspondence between sex and gender. It is curious that in German nouns can have three genders: masculine, feminine, and neuter, but the noun *Mädchen* 'girl' is neuter, not feminine!

Thus, the semantic representation of the English sentence *The little girls see the red flower* it is the same as the one given above, despite the absence of gender in English nouns and adjectives. The representation of the corresponding Russian sentence is the same too, though the word used for *red* in Russian has masculine gender,

because of its agreement in gender with corresponding noun of masculine.[16]

Nevertheless, the cases when semantic representations for two or more utterances with seemingly the same meaning do occur. In such situations, linguists hope to find a universal representation via decomposition and even atomization of the meaning of several semantic components.

In natural sciences, such as physics, researchers usually try to divide all the entities under consideration into the simplest possible, i.e., atomic, or elementary, units and then to deduce properties of their conglomerations from the properties of these elementary entities. In principle, linguistics has the same objective. It tries to find the atomic elements of meaning usually called *semantic primitives*, or *semes*.

Semes are considered indefinable, since they cannot be interpreted in terms of any other linguistic meanings. Nevertheless, they can be explained to human readers by examples from the extralinguistic reality, such as pictures, sound records, videos, etc. All other components of semantic representation should be then expressed through the semes.

In other words, each predicate or its terms can be usually represented in the semantic representation of text in a more detailed manner, such as a logical formula or a semantic graph. For example, we can decompose

$$MATAR(x) \rightarrow CAUSAR(MORIR(x)) \rightarrow CAUSAR(CESAR(VIVIR(x))),$$

i.e., $MATAR(x)$ is something like '*causar cesar el vivir*(x),' or '*cause stop living*(x),' where the predicates $CESAR(x)$, $VIVIR(y)$, and $CAUSAR(z)$ are more elementary than the initial predicate $MATAR(x)$.[17]

[16] In Russian: *Malen'kie devochki vidjat krasnyj cvetok*; the two last words are singular masculine.

[17] Some researchers consider the predicate *TO LIVE*(x) elementary.

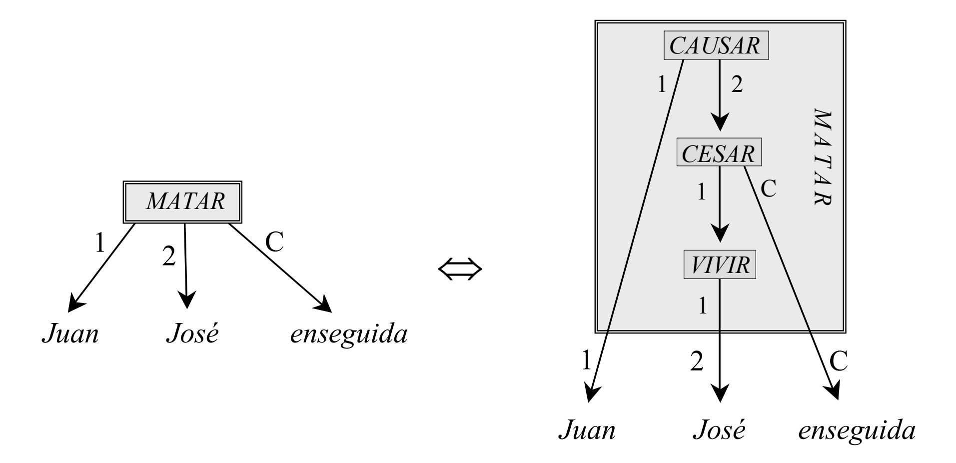


FIGURE IV.9. *Decomposition of the verb* MATAR *into semes.*

Figure IV.9 shows a decomposition of the sentence *Juan mató a José enseguida* = *Juan causó a José cesar vivir enseguida* in the mentioned more primitive notions. Note that the number labels of valencies of the whole combination of the primitives can differ from the number labels of corresponding valencies of the member primitives: e.g., the actant 2 of the whole combination is the actant 1 of the component *VIVIR*. The mark C in Figure IV.9 stands for the circumstantial relation (which is not a valency but something inverse, i.e., a passive semantic valency).

Over the past 30 years, ambitious attempts to find and describe a limited number of semes, to which a major part of the semantics of a natural language would be reduced, have not been successful.

Some scientists agree that the expected number of such semes is not much more than 2´000, but until now, this figure is still debatable. To comply with needs of computational linguistics, everybody agreed that it is sufficient to disintegrate meanings of lexemes to a reasonable limit implied by the application.

Therefore, computational linguistics uses many evidently non-elementary terms and logical predicates in the semantic representation. From this point of view, the translation from one cognate language to another does not need any disintegration of meaning at all.

Once again, only practical results help computational linguists to judge what meaning representation is the best for the selected application domain.

NOT-UNIQUENESS OF MEANING ⇒ TEXT MAPPING: SYNONYMY

Returning to the mapping of Meanings to Texts and vice versa, we should mention that, in contrast to common mathematical functions, this mapping is not unique in both directions, i.e., it is of the many-to-many type. In this section, we will discuss one direction of the mapping: from Meanings to Texts.

Different texts or their fragments can be, in the opinion of all or the majority of people, equivalent in their meanings. In other words, two or more texts can be mapped to the same element of the set of Meanings. In Figure IV.4, the Meaning M is represented with three different Texts T, i.e., these three Texts have the same Meaning.[18]

For example, the Spanish adjectives *pequeño* and *chico* are equivalent in many contexts, as well as the English words *small* and *little*. Such equivalent words are called *synonymous words*, or *synonyms*, and the phenomenon is called *synonymy* of words. We can consider also synonymy of word combinations (phrases) or sentences as well. In these cases the term *synonymous expressions* is used.

The words equivalent in all possible contexts are called *absolute synonyms*. Trivial examples of absolute synonymy are abbreviated and complete names of organizations, e.g. in Spanish *ONU* ≡ *Organización de las Naciones Unidas*. Nontrivial examples of absolute synonymy of single words are rather rare in any language. Examples from Mexican Spanish are: *alzadura* ≡ *alzamiento*, *acotación* ≡ *acotamiento*, *coche* ≡ *carro*.

However, it is more frequent that the two synonyms are equivalent in their meanings in many contexts, but not all.

[18] This is shown by the three bold arrows in Figure IV.4.

Sometimes the set of possible contexts for one such synonym covers the whole set of contexts for another synonym; this is called *inclusive synonymy*. Spanish examples are *querer* > *desear* > *anhelar*: *querer* is less specific than *desear* which in turn is less specific than *anhelar*. It means that in nearly every context we can substitute *desear* or *querer* for *anhelar*, but not in every context *anhelar* can be substituted for *querer* or *desear*.

Most frequently, though, we can find only some—perhaps significant—intersection of the possible sets of contexts. For example, the Spanish nouns *deseo* and *voluntad* are exchangeable in many cases, but in some cases only one of them can be used.

Such *partial synonyms* never have quite the same meaning. In some contexts, the difference is not relevant, so that they both can be used, whereas in other contexts the difference does not permit to replace one partial synonym with the other.

The book [24] is a typical dictionary of synonyms in printed form. The menu item *Language | Synonyms* in Microsoft Word is a typical example of an electronic dictionary of synonyms. However, many of the words that it contains in partial lists are not really synonyms, but related words, or partial synonyms, with a rather small intersection of common contexts.

NOT-UNIQUENESS OF TEXT ⇒ MEANING MAPPING: HOMONYMY

In the opposite direction—Texts to Meanings—a text or its fragment can exhibit two or more different meanings. That is, one element of the surface edge of the mapping (i.e. text) can correspond to two or more elements of the deep edge. We have already discussed this phenomenon in the section on automatic translation, where the example of Spanish word *gato* was given (see page 72). Many such examples can be found in any Spanish-English dictionary. A few more examples from Spanish are given below.

- The Spanish adjective *real* has two quite different meanings corresponding to the English *real* and *royal*.

- The Spanish verb *querer* has three different meanings corresponding to English *to wish*, *to like*, and *to love*.
- The Spanish noun *antigüedad* has three different meanings:
  - 'antiquity', i.e. a thing belonging to an ancient epoch,
  - 'antique', i.e. a memorial of classic antiquity,
  - 'seniority', i.e. length of period of service in years.

The words with the same textual representation but different meanings are called *homonymous* words, or *homonyms,* with respect to each other, and the phenomenon itself is called *homonymy*. Larger fragments of texts—such as word combinations (phrases) or sentences—can also be homonymous. Then the term *homonymous expressions* is used.

To explain the phenomenon of homonymy in more detail, we should resort again to the strict terms *lexeme* and *wordform*, rather than to the vague term *word*. Then we can distinguish the following important cases of homonymy:

- *Lexico-morphologic homonymy*: two wordforms belong to two different lexemes. This is the most general case of homonymy. For example, the string *aviso* is the wordform of both the verb *AVISAR* and the noun *AVISO*. The wordform *clasificación* belong to both the lexeme *CLASIFICACIÓN*$_1$ 'process of classification' and the lexeme *CLASIFICACIÓN*$_2$ 'result of classification,' though the wordform *clasificaciones* belongs only to *CLASIFICACIÓN*$_2$, since *CLASIFICACIÓN*$_1$ does not have the plural form. It should be noted that it is not relevant whether the name of the lexeme coincides with the specific homonymous wordform or not.

  Another case of lexico-morphologic homonymy is represented by two different lexemes whose sets of wordforms intersect in more than one wordforms. For example, the lexemes *RODAR* and *RUEDA* cover two homonymous wordforms, *rueda* and *ruedas*; the lexemes *IR* and *SER* have a number of wordforms in common: *fui*, *fuiste*, ..., *fueron*.

- Purely *lexical homonymy*: two or more lexemes have the same sets of wordforms, like Spanish *REAL*$_1$ 'real' and *REAL*$_2$ 'royal' (the both have the same wordform set {*real*, *reales*}) or *QUERER*$_1$ 'to wish,' *QUERER*$_2$ 'to like,' and *QUERER*$_3$ 'to love.'

- *Morpho-syntactic homonymy*: the whole sets of wordforms are the same for two or more lexemes, but these lexemes differ in meaning and in one or more morpho-syntactic properties. For example, Spanish lexemes (*el*) *frente* 'front' and (*la*) *frente* 'forehead' differ, in addition to meaning, in gender, which influences syntactical properties of the lexemes.

- Purely *morphologic homonymy*: two or more wordforms are different members of the wordform set for the same lexeme. For example, *fáciles* is the wordform for both masculine plural and feminine plural of the Spanish adjective *FÁCIL* 'easy.' We should admit this type of homonymy, since wordforms of Spanish adjectives generally differ in gender (e.g., *nuevos*, *nuevas* 'new').

Resolution of all these kinds of homonymy is performed by the human listener or reader according to the context of the wordform or based on the extralinguistic situation in which this form is used. In general, the reader or listener does not even take notice of any ambiguity. The corresponding mental operations are immediate and very effective. However, resolution of such ambiguity by computer requires sophisticated methods.

In common opinion, the resolution of homonymy (and ambiguity in general) is one of *the most difficult problems* of computational linguistics and must be dealt with as an essential and integral part of the language-understanding process.

Without automatic homonymy resolution, all the attempts to automatically "understand" natural language will be highly error-prone and have rather limited utility.

MORE ON HOMONYMY

In the field of computational linguistics, homonymous lexemes usually form separate entries in dictionaries. Linguistic analyzers must resolve the homonymy automatically, by choosing the correct option among those described in the dictionary.

For formal distinguishing of homonyms, their description in conventional dictionaries is usually divided into several subentries. The names of lexical homonyms are supplied with the indices (numbers) attached to the words in their standard dictionary form, just as we do it in this book. Of course, in text generation, when the program compiles a text containing such words, the indices are eliminated.

The purely lexical homonymy is maybe the most difficult to resolve since at the morphologic stage of text processing it is impossible to determine what homonym is true in this context. Since morphologic considerations are useless, it is necessary to process the hypotheses about several homonyms in parallel.

Concerning similarity of meaning of different lexical homonyms, various situations can be observed in any language. In some cases, such homonyms have no elements of meaning in common at all, like the Spanish $REAL_1$ 'real' and $REAL_2$ 'royal.' In other cases, the intersection of meaning is obvious, like in $QUERER_2$ 'to like' and $QUERER_3$ 'to love,' or $CLASIFICACIÓN_1$ 'process of classification' and $CLASIFICACIÓN_2$ 'result of classification.' In the latter cases, the relation can be exposed through the decomposition of meanings of the homonyms lexemes. The cases in which meanings intersect are referred to in general linguistics as *polysemy*.

For theoretical purposes, we can refer the whole set of homonymous lexemes connected in their meaning as *vocable*. For example, we may introduce the vocable $\{QUERER_1, QUERER_2, QUERER_3\}$. Or else we can take united lexeme, which is called *polysemic* one.

In computational linguistics, the intricate semantic structures of various lexemes are usually ignored. Thus, similarity in meaning is ignored too.

Nevertheless, for purely technical purposes, sets of any homonymous lexemes, no matter whether they are connected in meaning or not, can be considered. They might be referred as *pseudo-vocables*. For example, the pseudo-vocable *REAL* = $\{REAL_1, REAL_2\}$ can be introduced.

A more versatile approach to handle polysemy in computational linguistics has been developed in recent years using object-oriented ideas. Polysemic lexemes are represented as one superclass that reflects the common part of their meaning, and a number of subclasses then reflect their semantic differences.

A serious complication for computational linguistics is that new senses of old words are constantly being created in natural language. The older words are used in new meanings, for new situations or in new contexts. It has been observed that natural language has the property of self-enrichment and thus is very *productive*.

The ways of the enrichment of language are rather numerous, and the main of them are the following:

- A former lexeme is used in a *metaphorical* way. For example, numerous nouns denoting a process are used in many languages to refer also to a result of this process (*cf*. Spanish *declaración, publicación, interpretación*, etc.). The semantic proximity is thus exploited. For another example, the Spanish word *estética* 'esthetics' rather recently has acquired the meaning of heir-dressing saloon in Mexico. Since highly professional heir dressing really achieves esthetic goals, the semantic proximity is also evident here. The problem of resolution of metaphorical homonymy has been a topic of much research [51].

- A former lexeme is used in a *metonymical* way. Some proximity in place, form, configuration, function, or situation is used for metonymy. As the first example, the Spanish words *lentes* 'lenses,' *espejuelos* 'glasses,' and *gafas* 'hooks' are used in the meaning 'spectacles.' Thus, a part of a thing gives the name to the whole thing. As the second example, in many languages the name of an organization with a stable residence can be used to

designate its seat. For another example, *Ha llegado a la universidad* means that the person arrived at the building or the campus of the university. As the third example, the Spanish word *pluma* 'feather' is used also as 'pen.' As not back ago as in the middle of ninth century, feathers were used for writing, and then the newly invented tool for writing had kept by several languages as the name of its functional predecessor.

- A new lexeme is loaned from a foreign language. Meantime, the former, "native," lexeme can remain in the language, with essentially the same meaning. For example, English had adopted the Russian word *sputnik* in 1957, but the term *artificial satellite* is used as before.

- Commonly used abbreviations became common words, loosing their marking by uppercase letters. For example, the Spanish words *sida* and *ovni* are used now more frequently, then their synonymous counterparts *síndrome de inmunodeficiencia adquirida* and *objeto volante no identificado*.

One can see that metaphors, metonymies, loans, and former abbreviations broaden both homonymy and synonymy of language.

Returning to the description of all possible senses of homonymous words, we should admit that this problem does not have an agreed solution in lexicography. This can be proved by comparison of any two large dictionaries. Below, given are two entries with the same Spanish lexeme *estante* 'rack/bookcase/shelf,' one taken from the Dictionary of Anaya group [22] and the other from the Dictionary of Royal Academy of Spain (DRAE) [23].

estante (in Anaya Dictionary)

1. *m.* Armario sin puertas y con baldas.
2. *m.* Balda, anaquel.
3. *m.* Cada uno de los pies que sostienen la armadura de algunas máquinas.
4. *adj*. Parado, inmóvil.

estante (in DRAE)
1. *a. p. us*. de *estar*. Que está presente o permanente en un lugar. *Pedro, ESTANTE en la corte romana*.
2. *adj*. Aplícase al ganado, en especial lanar, que pasta constantemente dentro del término jurisdiccional en que está amillarado.
3. Dícese del ganadero o dueño de este ganado.
4. Mueble con anaqueles o entrepaños, y generalmente sin puertas, que sirve para colocar libros, papeles u otras cosas.
5. Anaquel.
6. Cada uno de los cuatro pies derechos que sostienen la armadura del batán, en que juegan los mazos.
7. Cada uno de los dos pies derechos sobre que se apoya y gira el eje horizontal de un torno.
8. *Murc*. El que en compañía de otros lleva los pasos en las procesiones de Semana Santa.
9. *Amér*. Cada uno de los maderos incorruptibles que, hincados en el suelo, sirven de sostén al armazón de las casas en las ciudades tropicales.
10. *Mar*. Palo o madero que se ponía sobre las mesas de guarnición para atar en él los aparejos de la nave.

One does not need to know Spanish to realize that the examples of the divergence in these two descriptions are numerous.

Some homonyms in a given language are translated into another language by non-homonymous lexemes, like the Spanish *antigüedad*.

In other cases, a set of homonyms in a given language is translated into a similar set of homonyms in the other language, like the Spanish *plato* when translated into the English *dish* (two possible interpretations are ‘portion of food’ and ‘kind of crockery’).

Thus, bilingual considerations sometimes help to find homonyms and distinguishing their meanings, though the main considerations should be deserved to the inner facts of the given language.

It can be concluded that synonymy and homonymy are important and unavoidable properties of any natural language. They bring

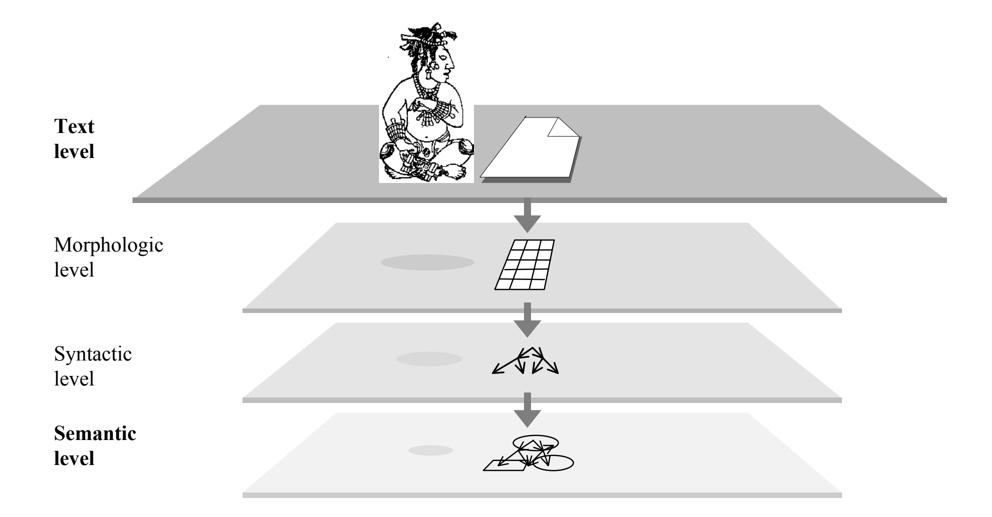


FIGURE IV.10. *Levels of linguistic representation.*

many heavy problems into computational linguistics, especially homonymy.

Classical lexicography can help to define these problems, but their resolution during the analysis is on computational linguistics.

MULTISTAGE CHARACTER OF THE MEANING ⇔ TEXT TRANSFORMER

The ambiguity of the Meaning ⇔ Text mapping in both directions, as well as the complicated structure of entities on both ends of the Meaning ⇔ Text transformer make it impossible to study this transformer without dividing the process of transformation into several sequential stages.

Existence of such stages in natural language is acknowledged by many linguists. In this way, intermediate levels of representation of the information under processing are introduced (see Figure IV.10), as well as partial transformers for transition from a level to an adjacent (see Figure IV.11).

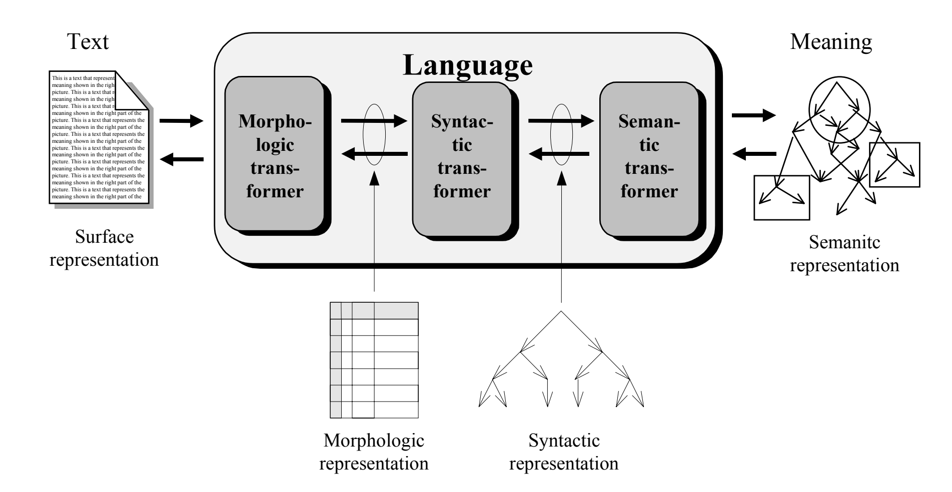


FIGURE IV.11. *Stages of transformation.*

Two intermediate levels are commonly accepted by all linguists, with small differences in the definitions, namely the morphologic and syntactic ones.

In fact, classical general linguistics laid the basis for such a division before any modern research. We will study these levels later in detail.

Other intermediate levels are often introduced by some linguists so that the partial transformers themselves are divided into sub-transformers. For example, a surface and a deep syntactic level are introduced for the syntactic stage, and deep and a surface morphologic level for the morphologic one.

Thus, we can imagine language as a multistage, or multilevel, Meaning ⇔ Text transformer (see Figure IV.12).

The knowledge necessary for each level of transformation is represented in computer dictionaries and computer grammars (see Figure IV.13). A *computer dictionary* is a collection of information on each word, and thus it is the main knowledge base of a text processing system. A *computer grammar* is a set of rules based on common properties of large groups of words. Hence, the grammar rules are equally applicable to many words.

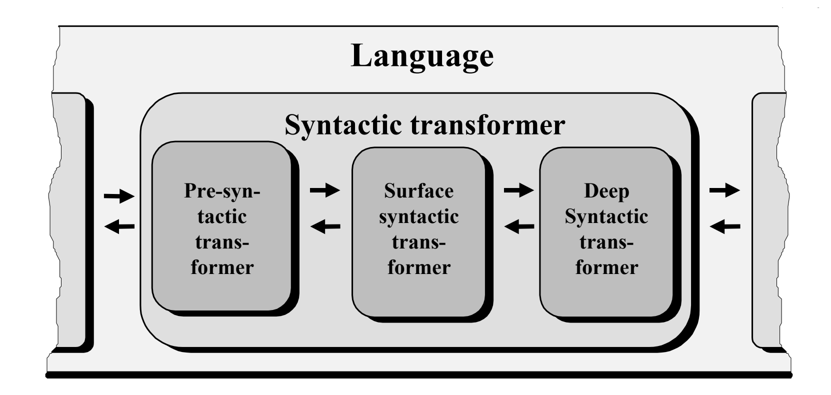


FIGURE IV.12. *Interlevel processing.*

Since the information stored in the dictionaries for each lexeme is specified for each linguistic level separately, program developers often distinguish a morphologic dictionary that specifies the morphologic information for each word, a syntactic dictionary, and a semantic dictionary, as in Figure IV.13.

In contrast, all information can be represented in one dictionary, giving for each lexeme all the necessary data. In this case, the dictionary entry for each lexeme has several *zones* that give the properties of this lexeme at the given linguistic level, i.e., a morphologic zone, syntactic zone, and semantic zone.

Clearly, these two representations of the dictionary are logically equivalent.

According to Figure IV.13, the information about lexemes is distributed among several linguistic levels. In Text, there are only wordforms. In analysis, lexemes as units under processing are involved at morphologic level. Then they are used at surface and deep syntactical levels and at last disappeared at semantic level, giving up their places to semantic elements. The latter elements conserve the meaning of lexemes, but are devoid of their purely grammatical properties, such as part of speech or gender. Hence, we can conclude that there is no level in the Text ⇒ Meaning transformer, which could be called lexical.

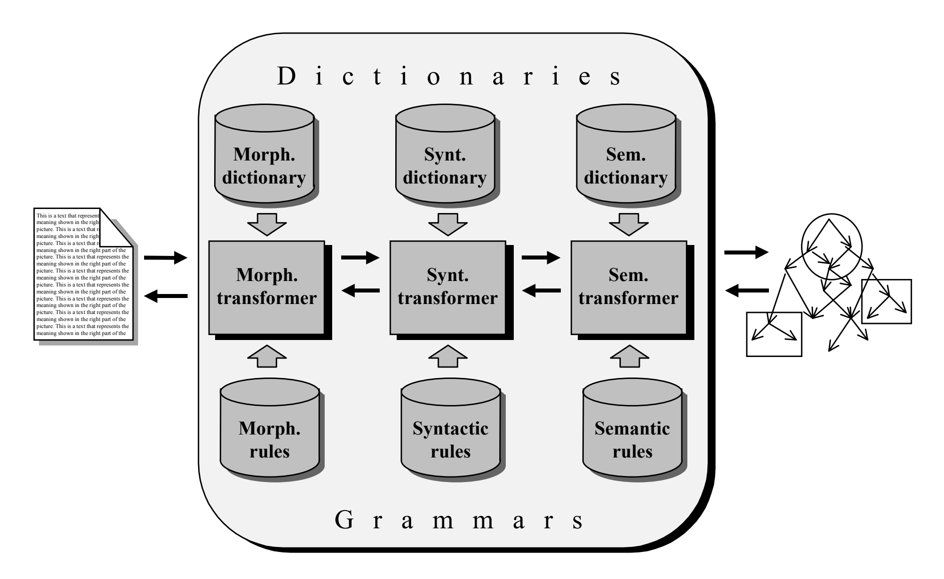


FIGURE IV.13. *The role of dictionaries and grammars in linguistic transformations.*

TRANSLATION AS A MULTISTAGE TRANSFORMATION

The task of translation from one natural language to another is a good illustration of multistage transformation of linguistic information.

Suppose there is a text in a language *A* that is to be translated into language *B*. As we have already argued, word-by-word translation leads to very poor and useless results. To translate the text with highest possible quality, the following stages of transformation are necessary:

- *First stage of analysis* starts from the source text in the language *A* and gives its morphologic representation specific for language *A*.

- *Second stage of analysis* starts from the morphologic representation and gives the syntactic representation specific for language *A*.
- *Third stage of analysis* starts from the syntactic representation and gives some level of semantic representation. The latter can be somewhat specific to language *A*, i.e., not universal, so that additional intra-level operations of "universalization" of semantic representation may be necessary.

The problem is that currently it is still not possible to reach the true semantic representation, i.e., the true level of Meaning, consisting of the universal and thus standard set of semes. Therefore, all practical systems have to stop this series of transformations at some level, as deep as possible, but not yet at that of universal Meaning.

- *The transfer stage* replaces the labels, i.e., of the conventional names of the concepts in language *A,* to the corresponding labels of language *B*. The result is the corresponding quasi-semantic level of representation in language *B*. In some cases, additional, more complex intra-level operations of "localization" are necessary at this stage.
- *First stage of synthesis* starts from the quasi-semantic representation with few features specific for the language *B*, and gives the syntactic representation quite specific for this language.
- *Second stage of synthesis* starts from the syntactic representation, and gives the morphologic representation specific for language *B*.
- *Third stage of synthesis* starts from the morphologic representation, and gives the target text in language *B*.

In the initial stages, the transformations go to the deep levels of the language, and then, in the last stages, return to the surface, with the ultimate result in textual form once more. The deeper the level reached, the smaller the difference between the representations of this level in both languages *A* and *B*. At the level of Meaning, there is no difference at all (except maybe for the labels at semes). The

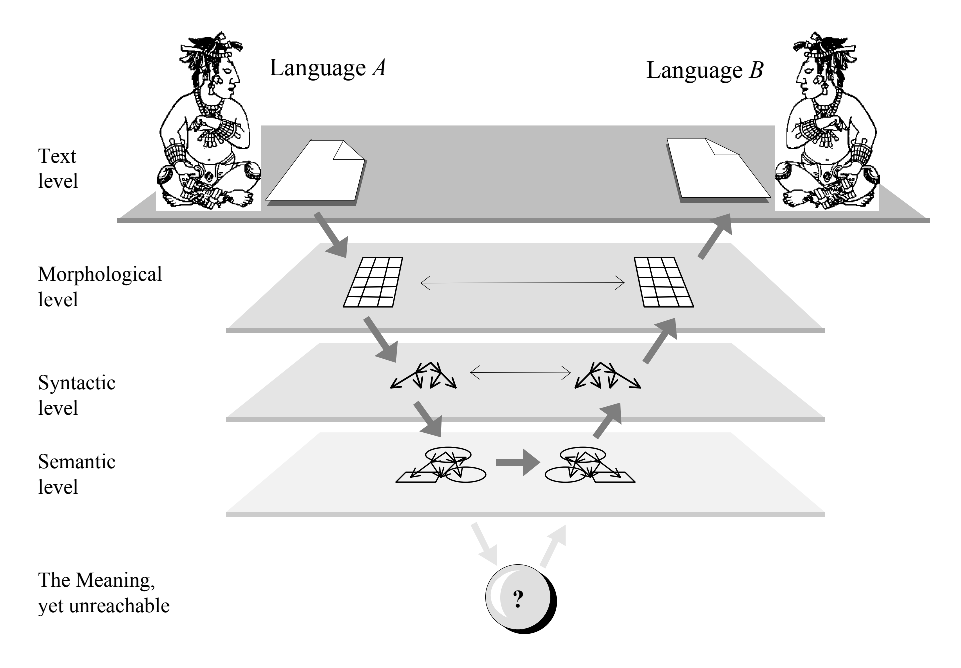


FIGURE IV.14. *Translation as multistage transformation.*

deeper the level reached during the transformations, the smaller the differences that have to be ignored, and the better the quality of translation (see Figure IV.14). This scheme shows only the general idea of all these representations.

The given scheme works for an arbitrary pair of natural languages. However, if the two languages are very similar in their structure, the deeper stages of the transformation might not be necessary.

For example, if we translate from Spanish into Portuguese, then, because these two languages differ mainly in their lexicon, it can be sufficient to use only the first stage of analysis and the last stage of synthesis, just replacing each Spanish word by the corresponding Portuguese one on the morphologic level.

In Figure IV.14 this would correspond then to the “horizontal” transition directly on this level.

TWO SIDES OF A SIGN

The notion of *sign,* so important for linguistics, was first proposed in a science called *semiotics*. The sign was defined as an entity consisting of two components, the *signifier* and the *signified*. Let us first consider some examples of *non-linguistic* signs taken from everyday life.

- If you see a picture with a stylized image of a man in a wheelchair on the wall in the subway, you know that the place under the image is intended for disabled people. This is a typical case of a sign: the picture itself, i.e., a specific figure in contrasting colors, is the signifier, while the suggestion to assist the handicapped persons is the signified.
- Twenty Mexican pesos have two equally possible signifiers: a coin with a portrait of Octavio Paz and a piece of paper with a portrait of Benito Juárez. The signified of both of them is the value of twenty pesos. Clearly, neither of them *has* this value, but instead they *denote* it. Thus, they are two different but synonymous signs.
- Raising one's hand (this gesture is the signifier) can have several different signifieds, for instance: (1) an intention to answer the teacher's question in a classroom, (2) a vote for or against a proposal at a meeting, (3) an intention to call a taxi in the street, etc. These are three different, though homonymous, signs.

LINGUISTIC SIGN

The notion of *linguistic sign* was introduced by Ferdinand de Saussure. By linguistic signs, we mean the entities used in natural languages, such as morphs, lexemes, and phrases.

Linguistic signs have several specific properties, the most obvious of which is that they are to be combined together into larger

signs and each one can in turn consist of several smaller signs. Natural language can be viewed as a system of linguistic signs.

As another property of linguistic sign, its signifier at the surface level consists of elementary parts, phonetic symbols in the phonetic transcription or letters in the written form of language. These parts do not have any signified of their own: a letter has no meaning, but certain strings of letters do have it.

We have already mentioned other notation systems to represent words of natural language, such as hieroglyphic writing. Each hieroglyph usually has its own meaning, so a hieroglyph *is* a linguistic sign. The gestures in the sign language for deaf people in some cases do have there own meanings, like hieroglyphs, and in other cases do not have any meaning of their own and serve like letters.

LINGUISTIC SIGN IN THE MMT

In addition to the two well-known components of a sign, in the Meaning ⇔ Text Theory yet another, a third component of a sign, is considered essential: a record about its ability or inability to combine with other specific signs. This additional component is called *syntactics* of the linguistic sign. For example, the ending morph -*ar* for Spanish infinitives has the same signified as -*er* and -*ir*. However, only one of them can be used in the wordform *hablar*, due to the syntactics of these three endings, as well as to the syntactics of the stem *habl*-. We can imagine a linguistic sign as it is shown on Figure IV.15.

Thus, syntactics of linguistic signs helps to choose one correct sign from a set of synonymous signs, to be used in a specific context. For example, we choose the ending -*ar* for the stem *habl*- since it does accept just it. Beyond that, syntactics of linguistic signs helps us to disambiguate the signs, i.e., to decide which of the homonymous signs is used in a specific context.

For the extralinguistic example given above, the classroom is the context, in which we interpret the raising of one's hand as the intention to say something rather than to call a taxi.

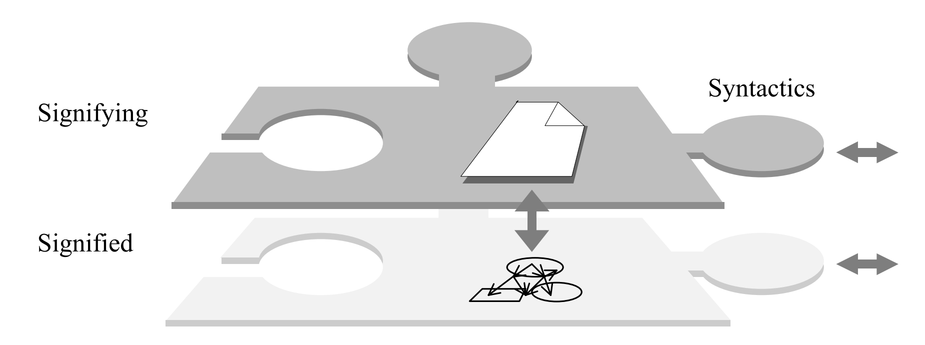


FIGURE IV.15. *Syntactics of a linguistic sign.*

Similarly, the presence of the Spanish stem *salt*- is sufficient to interpret the -*as* ending as present tense, second person, singular in *saltas*, rather than feminine, plural as in the presence of the stem *roj*-: *rojas*.

LINGUISTIC SIGN IN HPSG

In Head-driven Phrase Structure Grammar a linguistic sign, as usually, consists of two main components, a signifier and a signified. The signifier is defined as a phoneme string (or a sequence of such strings). Taking into account the correspondence between acoustic and written forms of language, such signifier can be identified as conceptually coinciding with elements of Text in the MTT.

As to the signified, an object of special type SYNSEM is introduced for it. An object of this type is a structure (namely, a labeled tree called *feature structure*) with arcs representing features of various linguistic levels: morphologic, syntactic, and semantic, in a mixture. For a minimal element of the text, i.e., for a wordform, these features show:

- How to combine this wordform with the other wordforms in the context when forming syntactic structures?
- What logical predicate this word can be a part of?
- What role this word can play in this predicate?

Simple considerations show that SYNSEM in HPSG unites the properties of Meaning and syntactics of a sign as defined in the framework of the MTT, i.e., SYNSEM covers syntactics plus semantics. Hence, if all relevant linguistic facts are taken into consideration equally and properly by the two approaches, both definitions of the linguistic sign, in HPSG and MTT, should lead to the same results.

ARE SIGNIFIERS GIVEN BY NATURE OR BY CONVENTION?

The notion of sign appeared rather recently. However, the notions equivalent to the signifier and the signified were discussed in science from the times of the ancient Greeks. For several centuries, it has been debated whether the signifiers of things are given to us by nature or by human convention.

The proponents of the first point of view argued that some words in any language directly arose from *onomatopoeia*, or sound imitation. Indeed, we denote the sounds produced by a cat with verb *mew* or *meow* in English, *maullar* in Spanish, and *miaukat'* in Russian. Hence, according to them, common signifiers lead to similar signifieds, thus creating a material link between the two aspects of the sign.

The opponents of this view objected that only a minority of words in any language takes their origin from such imitation, all the other words being quite arbitrary. For example, there is no resemblance between Spanish *pared*, English *wall*, and Russian *stena*, though all of them signify the same entity. Meanwhile, the phonetic similarity of the German *Wand* to the English *wall*, or the French *paroi* to the Spanish *pared*, or the Bulgarian *stena* to the Russian *stena* are caused by purely historic reasons, i.e., by the common origin of those pairs of languages.

The latter point of view has become prevalent, and now nobody looks for material links between the two sides of linguistic signs. Thus, a linguistic sign is considered a human convention that assigns specific meanings to some material things such as strings of

letters, sequences of sounds, pictures in hieroglyphic writing or gestures in sign language.

GENERATIVE, MTT, AND CONSTRAINT IDEAS IN COMPARISON

In this book, three major approaches to linguistic description have been discussed till now, with different degree of detail: (1) generative approach developed by N. Chomsky, (2) the Meaning ⇔ Text approach developed by I. Mel'čuk, and (3) constraint-based approach exemplified by the HPSG theory. In the ideal case, they produce equivalent results on identical language inputs. However, they have deep differences in the underlying ideas. In addition, they use similar terminology, but with different meaning, which may be misleading. In this section, we will compare their underlying ideas and the terminology. To make so different paradigms comparable, we will take only a bird's-eye view of them, emphasizing the crucial commonalities and differences, but in no way pretending to a more deep description of any of these approaches just now.

Perhaps the most important commonality of the three approaches is that they can be viewed in terms of linguistic signs. All of them describe the structure of the signs of the given language. All of them are used in computational practice to find the Meaning corresponding to a given Text and vice versa. However, the way they describe the signs of language, and as a consequence the way those descriptions are used to build computer programs, is different. *Generative idea*. The initial motivation for the generative idea was the fact that describing the language is much more difficult, labor-consuming, and error-prone task than writing a program that uses such a description for text analysis. Thus, the formalism for description of language should be oriented to the process of describing and not to the process of practical application. Once created, such a description can be applied somehow.

Now, what is to describe a given language? In the view of generative tradition, it means, roughly speaking, to list all signs in it (in fact, this is frequently referred to as generative idea). Clearly, for a

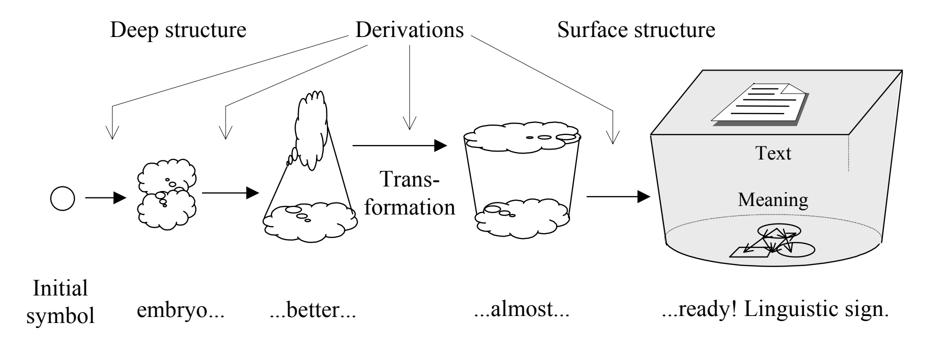


FIGURE IV.16. *Generative idea.*

natural language it is impossible to literally list all signs in it, since their number is infinite. Thus, more strictly speaking, a generative grammar describes an algorithm that lists only the correct signs of the given language, and lists them all—in the sense that any given sign would appear in its output after a some time, perhaps long enough. The very name *generative* grammar is due to that it describes the process of *generating* all language signs, one by one at a time.

There can be many ways to generate language signs. The specific kind of generative grammars suggested by N. Chomsky constructs each sign gradually, through a series of intermediate, half-finished sign "embryos" of different degree of maturity (see Figure IV.16). All of them are built starting from the same "egg-cell" called *initial symbol*, which is not a sign of the given language. A very simple example of the rules for such gradual building is given on the pages 35 to 39; in this example, the tree structure can be roughly considered the Meaning of the corresponding string.

Where the infinity of generated signs comes from? At each step, called *derivation*, the generation can be continued in different ways, with any number of derivation steps. Thus, there exist an infinite number of signs with very long derivation paths, though for each specific sign its derivation process is finite.

However, all this generation process is only imaginable, and serves for the formalism of description of language. It is not—and is

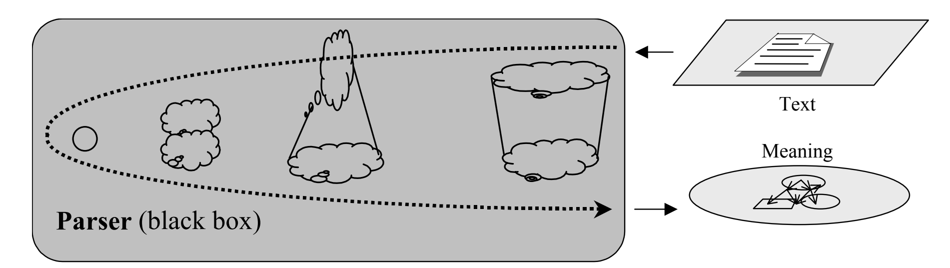


FIGURE IV.17. *Practical application of the generative idea.*

not intended to be—applied in practice for the generation of an infinitely long list of language expressions, which would be senseless. The use of the description—once created—for passing from Text to Meaning and vice versa is indirect. A program called *parser* is developed by a mathematician (not a linguist) by means of automatic "reversing" of the original description of the generative process.

This program can answer the questions: *What signs would it generate that have the given Text as the signifier? What signs would it generate that have the given Meaning as signified?* (See Figure IV.17.)

The parser does not really try to generate any signs, but instead solves such an equation using the data structures and algorithms quite different from the original description of the generating process.

The result produced by such a black box is, however, exactly the same: given a Text, the parser finds such Meaning that the corresponding sign belongs to the given language, i.e., *would* be generated by the imaginable generation algorithm. However, the description of the imaginable generation process is much clearer than the description of the internal structures automatically built by the parser for the practical applications.

Meaning ⇔ Text idea. As any other grammar, it is aimed at the practical application in language analysis and synthesis. Unlike generative grammar, it does not concentrate on enumeration of all possible language signs, but instead on the laws of the correspondence between the Text and the Meaning in any sign of the given

language. Whereas for a given text, a generative grammar can answer the question *Do any signs with such Text exist, and if so, what are their Meanings?*, a the MTT grammar only guarantees the answer to the question *If signs with such Text existed, what would be their Meanings?*

In practice, the MTT models usually *can* distinguish existing signs from ungrammatical ones, but mainly as a side effect. This makes the MTT models more robust in parsing.

Another idea underlying the MTT approach is that linguists are good enough at the intuitive understanding of the correspondence between Texts and Meanings, and can describe such correspondences directly. This allows avoiding the complications of generative grammars concerning the reversion of rules. Instead, the rules are applied to the corresponding data structures directly as written down by the linguist (such property of a grammar is sometimes called *type transparency* [47]). Direct application of the rules greatly simplifies debugging of the grammar. In addition, the direct description of the correspondence between Text and Meaning is supposed to better suite the linguistic reality and thus results in less number of rules.

Similarly to the situation with generative grammars, there can be many ways to describe the correspondence between Text and Meaning. The specific kind of the MTT grammars suggested by I. Mel'čuk describes such a correspondence gradually, through many intermediate, half-finished almost-Meanings, half-Meanings, half-Texts, and almost-Texts, as if they were located inside the same sign between its Meaning and Text (see Figure IV.18).

Since the MTT and the generative approach developed rather independently, by accident, they use similar terms in quite different and independent meanings. Below we explain the differences in the use of some terms, though these informal explanations are *not* strict definitions.

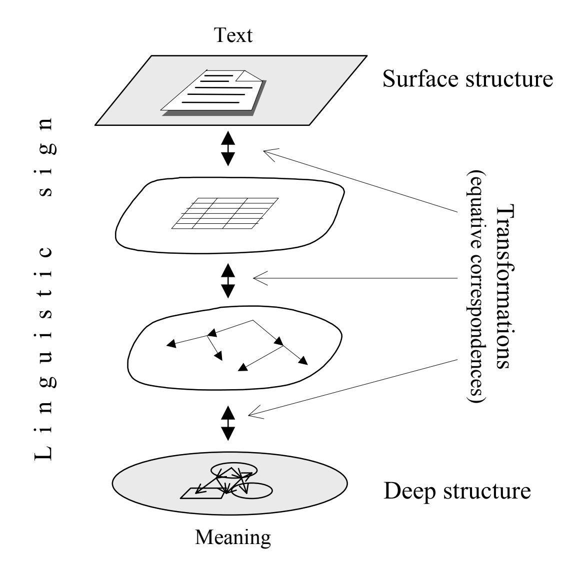


FIGURE IV.18. *Meaning ⇔ Text idea.*

- In generative grammar (see Figure IV.16):
  - *Transformation*: a term used in early works by N. Chomsky for a specific kind of non-context-free derivation.
  - *Deep structure*, in the transformational grammar, is a half-finished sign with a special structure to which a transformation is applied to obtain a "readier" sign. It is nearer to the initial symbol than the surface structure.
  - *Surface structure* is a half-finished sign obtained as the result of the transformation. It is nearer to the ready sign than the deep structure.
  - *Generation* is used roughly as a synonym of derivation, to refer to the process of enumeration of the signs in the given language.
- In the MTT (see Figure IV.18):
  - *Transformation* is sometimes used for equative correspondences between representations on different levels.

- *Deep structure* concerns the representation nearer to Meaning.
- *Surface structure* concerns the representation nearer to Text.
- *Generation* (of text) is used sometimes as a synonym of synthesis, i.e., construction of Text for the given Meaning.

Constraint-based idea. Similarly to the generative grammar, a constraint-based grammar describes what signs exist in the given language, however not by means of explicit listing (generation) of all such signs, but rather by stating the conditions (constraints) each sign of the given language must satisfy.

It can be viewed as if it specified what signs do *not* exist in the given language: if you remove one rule (generation option) from a generative grammar, it will generate *less* signs. If you remove one rule (constraint) from a constraint-based grammar, it will allow *more* signs (i.e., allow some signs that really are ungrammatical in the given language). Hence is the name *constraint-based*. (See also page 44.)

Since constraint-based grammars do not use the generation process shown on Figure IV.16, their rules are applied within the same sign rather than to obtain one sign from another (half-finished) one.

This makes it similar to the MTT. Indeed, though the constraint-based approach was originated in the generative tradition, modern constraint-based grammars such as HPSG show less and less similarities with Chomskian tradition and more and more similarity—not in the formalism but in meaningful linguistic structures—with the MTT.

A constraint-based grammar is like a system of equations. Let us consider a simple mathematical analogy.

Each sheet of this book is numbered at both sides. Consider the side with even numbers. Looking at the page number, say, 32, you can guess that it is printed on the 16-th sheet of the book. Let what you see be Text and what you guess be Meaning; then this page number corresponds to a "sign" <32, 16>, where we denote <T, M> a sign with the Text T and Meaning M. In order to describe such a

"language", the three approaches would use different mathematical constructions (of course, in a very rough analogy):

- Generative grammar is like a *recurrent formula*: The sign $<2, 1>$ (analogue of the initial symbol) belongs to this "language", and if $<x, y>$ belongs to it, then $<x + 2, y + 1>$ belongs to it (analogue of a generation rule). Note that some effort is needed to figure out from this description how to find a sheet number by a page number.

- The MTT grammar is like an *algorithm*: given the page number x, its sheet number is calculated as $x/2$; given a sheet number y, its page number is calculated as $2 \times y$. Note that we have made no attempt to describe dealing with, or excluding of, odd page numbers x, which in fact do not belong to our "language."

- Constraint-based grammar is like an *equation* or *system of equations*. Just those signs belong to our "language," for which $x = 2y$. Note that this description is the most elegant and simple, completely and accurately describes our "language," and requires less reversing effort for practical application than the first one. However, it is more complex than the second one.

Constraint-based idea is a very promising approach adopted by the majority of contemporaneous grammar formalisms. Probably with time, the linguistic findings of the MTT will be re-formulated in the form of constraint-based rules, possibly by a kind of merging of linguistic heritage of the MTT and formalisms developed in frame of HPSG. However, for the time being we consider the MTT more mature and thus richer in detailed description of a vast variety of linguistic phenomena. In addition, this approach is most directly applicable, i.e., it does not need any reversing.

As to the practical implementation of HPSG parsers, it is still an ongoing effort at present.

CONCLUSIONS

The definition of language has been suggested as a transformer between the two equivalent representations of information, the Text, i.e., the surface textual representation, and the Meaning, i.e., the deep semantic representation. This transformation is ambiguous in both directions: a homonymous Text corresponds to several different Meanings, and several synonymous Texts correspond to the same Meaning.

The description of the transformation process is greatly simplified by introducing intermediate levels of information representation, of which the main are morphologic and syntactic. At each level, some of the problems arising from synonymy and homonymy can be solved.

The general definitions of linguistic sign in Meaning ⇔ Text Theory and in Head-driven Phrase Structure Grammar turned out to be in essence equivalent.

V. LINGUISTIC MODELS

THROUGHOUT THE PREVIOUS CHAPTERS, you have learned, on the one hand, that for many computer applications, detailed linguistic knowledge is necessary and, on the other hand, that natural language has a sophisticated structure, which is not easy to represent.

Thus, any application needs a description of language, i.e., the knowledge about its main properties. Such knowledge is organized in a *model* of language. The structure and degree of detail depend on the application's needs.

Our objectives now are to discuss the problem of modeling in computational linguistics. We observe the modeling in general, describe shortly the neurolinguistic and psycholinguistic models, and then discuss the functional models of natural language, with a special emphasis on common features of these models.

WHAT IS MODELING IN GENERAL?

In natural sciences, we usually consider the system *A* to be a *model* of the system *B* if *A* is similar to *B* in some important properties and exhibits somewhat similar behavior in similar circumstances. According to this definition, it is unnecessary the physical nature of the modeling system *A* be the same as of the modeled system *B*. For example, *B* can be a technological aggregation in chemistry, while *A* can be a set of differential equations, i.e., a conceptual system of a mathematical nature. A mathematical model is usually the best if it really ensures sufficient approximation.

In the linguistic domain, we need a modeling system that, after receiving information of a linguistic nature in its input, exhibits in its output an acceptable degree of similarity to the results of natural language activity of the human brain. Such a definition follows the idea of Alan Turing mentioned above.

NEUROLINGUISTIC MODELS

Neurolinguistic models investigate the links between any external speech activity of human beings and the corresponding electrical and humoral activities of nerves in their brain.

It would be interesting to investigate in detail what part of the brain is activated when a human is preparing and producing an utterance or is trying to understand an utterance just heard. Unfortunately, the problem to discover the way people think while speaking or understanding is tremendously difficult. Indeed, the unique objective way for a researcher to reveal the ways of the human thinking by neurophysiological methods is to synchronously investigate electrical and humoral activities in a multiplicity of places in the human brain.

The brain of any human consists of a tremendous number of neural cells, or neurons. The neurons are not elementary units by themselves, since each of them is connected with hundreds of other neurons by their dendrites and axons in a very complicated manner.

The dimensions of neurons, dendrites, and axons are so small that the techniques of modern neurophysiology provide no practical opportunity to observe each neuron separately. What is more, the intrusion of observing tools inside the brain evidently changes the activity under observation. To understand in detail the internal functioning of the brain, it would be necessary to observe vast groups of neurons activated in the multistage process and trace back a grandiose multiplicity of causal chains.

Then, after this tracing, the researcher would be able to express the detected traces as rules and algorithms presumably used by the human brain. This is what might be called a neurolinguistic model of language processing. However, the extremely large number of possible combinations of signals traced in such a way is well beyond the capability of any computer to handle.

Leaving aside the inadequacy of the modern neurophysical techniques, the mathematical tools for such a modeling are insufficient either. We cannot even hope to get close to the information activity

of neuron groups in our brain with rather simple mathematical models. According to the modern point of view, the neurons are complicated elements of logical type. Thus, a modeling system should contain elements of the switching circuitry. However, neither deterministic nor stochastic theory of approximation by such a circuitry has been developed well nowadays.

About 30 years ago, neural networks were proposed as a tool of artificial intelligence research. They consist of elements referred to as *formal neurons*. These are standard logical computational units with rather limited abilities. There have been numerous studies attempting to apply neural networks to various tasks of artificial intelligence and particularly to language processing. If there are some statistical trends in phenomena under observation and modeling, these trends can be recognized and taken into account by this technique.

However, for computational linguistics the problem of how to choice the input and output signals for modeling of the brain activity remains unclear. If we propose some specific inner representations for these purposes, then the formal neurons will only model our inventions rather than real language processes.

For this reason, without revolutionary changes in the research techniques and new approaches for treatment of observable data, neurolinguistic models of natural language understanding are unlikely to give good results in the nearest future. At present, only very crude features of brain activity can be observed effectively, such as which areas within the brain show neuronal activity associated with human memory, general reasoning ability, and so forth.

PSYCHOLINGUISTIC MODELS

Psycholinguistics is a science investigating the speech activity of humans, including perception and forming of utterances, via psychological methods. After creating its hypotheses and models, psycholinguistics tests them through psychological experiments. There-

fore, psycholinguistics is similar to linguistics in its objects of investigation and similar to psychology in its methods.

Here is an example of a psycholinguistic experiment. The subject, i.e., the person under observation, is given, one by one, a sequence of specially selected words as input stimuli. Then the subject is prompted to give the response to each word as any other word first coming to his or her mind. The pairs "stimulus—response" are recorded: *rosa*—*roja*, *padre*—*madre*, *mano*—*pie*, etc. Based on such experiments, psycholinguists put forward a hypothesis that various psychological types of people demonstrate specific types of associative choices, so that they can give a true identification of the personality under investigation based exclusively on such tests.

In another experiment, the objectives can be nearer to linguistics as such. The series of utterances with some syntactic ambiguities are given to the human subjects. The time required to the person under test for disambiguation is measured, when he or she selects at last one of possible hypotheses. On the grounds of experimental statistics, a hypothesis is advanced how a human can understand such constructions and what is the humans' way for the disambiguation.

Psycholinguistics tries also to describe the teaching of native and not native language, social influence of a speech on humans, and so forth. In such a way, psycholinguistics aims to explain some purely psychological features of people concerning their speech behavior. Psychological features, in their turn, are closely related to the social behavior of a person. Two kinds of activities, both of very high complexity, are studied in parallel, and each aspect of the science exploits techniques and suggestions from the other.

Therefore, psycholinguistics usually does not have linguistic ideas of its own. It adopts them from various branches of linguistics and then uses them for its own purposes without much critical review and feedback. Purely psychological methods well adapted to the problems of linguistics as such have not yet been proposed, so that these methods give little to generalized (holistic) models of language.

FUNCTIONAL MODELS OF LANGUAGE

In terms of cybernetics, natural language is considered as a *black box* for the researcher. A black box is a device with observable input and output but with a completely unobservable inner structure. In the framework of this type of model, language is thought to be an imaginary "speaking device": the researcher asks the device some questions and records its answers.

The problem of the cybernetic modeling of natural language is more difficult than in other cases, since there are two such boxes, the analyzing and synthesizing ones, working in opposite directions. The analyzing block processes the utterances and the synthesizing block produces the reactions to them.

A researcher observes the input of the analyzing block and the output of the synthesizing block, and tries to reconstruct the inner structure of each block separately. Unfortunately, the output of the analyzer is not directly used as the input of the synthesizer. There is a block of reasoning in between, and its behavior is not described in linguistic terms, so that it is not so easy to recognize either.

The main method of linguistics is to construct a model of natural language, based on the observable input and output texts, and on the linguist's intuition, or introspection. The linguists analyze their own intuition, put forward hypotheses, build models and test them on additional linguistic material. In theoretical linguistics, the novel approaches can be tested against (compared with) intuitions of other linguists, while in computational linguistics these approaches can be also tested through various applications.

In this way, linguists have proposed *functional models* of language. These models are intended to give the rules of conversion of the input linguistic information to the output information, without any attempt to directly reproduce the internal mechanisms of brain activity. No anthropomorphic features of the processing steps are searched for, and no direct emulation of brain's responses is adduced. However, the ultimate *results* of all processing steps should be as near as possible to those of the human brain.

So far, functional models have proven to be the most successful linguistic models, probably because they are based on real data with conceivable structure, easily accessible and available in unlimited supply, namely, on texts and recorded speech.

RESEARCH LINGUISTIC MODELS

There are still other models of interest for linguistics. They are called *research models*. At input, they take texts in natural language, maybe prepared or formatted in a special manner beforehand. As an output, they produce other texts, usually strictly formatted and representing the contents of dictionaries, grammar tables, rules or anything similar to be used as a part of functional models.

As an example, we can collect all the agreed pairs like "article—noun" or "noun—adjective," or all the prepositions occurring in an open, i.e., not prepared, text in natural language. As another example, we can extract from the text of a dictionary those items of a given part of speech, which contain a predetermined combination of features.

Thus, research models are tools for constructing functional models. They simulate linguists in their research, whereas the functional models simulate humans in the speech producing and understanding.

COMMON FEATURES OF MODERN MODELS OF LANGUAGE

The modern models of language have turned out to possess several common features that are very important for the comprehension and use of these models. One of these models is given by the Meaning ⇔ Text Theory already mentioned. Another model is that based on the Head-Driven Phrase Structure Grammar. The Chomskian approach within the Western linguistic tradition includes various other models different from HPSG.

Here are the main common features of all these models:

- *Functionality of the model*. The linguistic models try to reproduce functions of language without directly reproducing the features of activity of the brain, which is the motor of human language.

- *Opposition of the textual/phonetic form of language to its semantic representation*. The manual [9], depicting three different well-known syntactic theories (including one of the recent variant of the theory by N. Chomsky), notices: "Language ultimately expresses a relation between sound at one end of the linguistic spectrum and meaning at the other." Just as the diffuse notion *spectrum* is somehow sharpened, we have the same definition of language as in the MTT. The outer, observable form of language activity is a text, i.e., strings of phonetic symbols or letters, whereas the inner, hidden form of the same information is the meaning of this text. Language relates two these forms of the same information.

- *Generalizing character of language*. Separate utterances, within a speech or a text, are considered not as the language, but as samples of its functioning. The language is a theoretical generalization of the open and hence infinite set of utterances. The generalization brings in features, types, structures, levels, rules, etc., which are not directly observable. Rather these theoretical constructs are fruits of linguist's intuition and are to be repeatedly tested on new utterances and the intuition of other linguists. The generalization feature is connected with the opposition *competence* vs. *performance* in Chomskian theory and to the much earlier opposition *language* vs. *speech* in the theory of Ferdinand de Saussure.

- *Dynamic character of the model*. A functional model does not only propose a set of linguistic notions, but also shows (by means of rules) how these notions are used in the processing of utterances.

- *Formal character of the model*. A functional model is a system of rules sufficiently rigorous to be applied to any text by a person or an automaton quite formally, without intervention of the model's author or anybody else. The application of the rules to the given text or the given meaning always produces the same result. Any part of a functional model can in principle be expressed in a strict mathematical form and thus algorithmized.[19] If no ready mathematical tool is available at present, a new tool should be created. The presupposed properties of recognizability and algorithmizability of natural language are very important for the linguistic models aimed at computer implementation.
- *Non-generative character of the model*. Information does not arise or generated within the model; it merely acquires a form corresponding to other linguistic level. We may thus call the correspondences between levels *equative correspondences*. On the contrary, in the original generative grammars by Chomsky, the strings of symbols that can be interpreted as utterances are generated from an initial symbol, which has just abstract sense of a sentence. As to transformations by Chomsky in their initial form, they may change the meaning of an utterance, and thus they were not equative correspondences.
- *Independence of the model from direction of transformation.* The description of a language is independent of the direction of linguistic processing. If the processing submits to some rules, these rules should be given in equative (i.e., preserving the meaning) bi-directional form, or else they should permit reversion in principle.
- *Independence of algorithms from data*. A description of language structures should be considered apart from algorithms using this description. Knowledge about language does not imply a specific type of algorithms. On the contrary, in many situations an algorithm implementing some rules can have numerous options. For

[19] Formality of any model is quite natural from the programming point of view.

example, the MTT describes the text level separately from the morphologic and syntactic levels of the representation of the same utterance. Nevertheless, one can imagine an algorithm of analysis that begins to construct the corresponding part of the syntactic representation just as the morphologic representation of the first word in the utterance is formed. In the cases when linguistic knowledge is presented in declarative form with the highest possible consistency, implementing algorithms proved to be rather universal, i.e., equally applicable to several languages. (Such linguistic universality has something in common with the Universal Grammar that N. Chomsky has claimed to create.) The analogous distinction between algorithms and data is used with great success in modern compilers of programming languages (cf. compiler-compilers).

- *Emphasis on detailed dictionaries*. The main part of the description of any language implies words of the language. Hence, dictionaries containing descriptions of separate words are considered the principal part of a rigorous language description. Only very general properties of vast classes and subclasses of lexemes are abstracted from dictionaries, in order to constitute formal grammars.

SPECIFIC FEATURES OF THE MEANING ⇔ TEXT MODEL

The Meaning ⇔ Text Model was selected for the most detailed study in these books, and it is necessary now to give a short synopsis of its specific features.

- *Orientation to synthesis*. With the announced equivalence of the directions of synthesis and analysis, the synthesis is considered primary and more important for linguistics. Synthesis uses the entire linguistic knowledge about the text to be produced, whereas analysis uses both purely linguistic and extralinguistic knowledge, would it be encyclopedic information about the

world or information about the current situation. That is why analysis is sometimes possible on the base of a partial linguistic knowledge. This can be illustrated by the fact that we sometimes can read a paper in a nearly unknown language, if the field and subject of the paper are well known to us. (We then heavily exploit our extralinguistic knowledge.) However, text analysis is considered more important for modern applications. That is why the generative grammar approach makes special emphasis on analysis, whereas for synthesis separate theories are proposed [49]. The Meaning ⇔ Text model admits a separate description for analysis, but postulates that it should contain the complete linguistic and any additional extralinguistic part.

- *Multilevel character of the model*. The model explicitly introduces an increased number of levels in language: textual, two morphologic (surface and deep), two syntactic (surface and deep), and semantic one. The representation of one level is considered equivalent to that of any other level. The equative Meaning ⇒ Text processor and the opposite Text ⇒ Meaning processor are broken into several partial modules converting data from one level to the adjacent one. Each intermediate level presents the output of one module and, at the same time, the input of another module. The division of the model in several modules must simplify rules of inter-level conversions.
- *Reinforced information-preserving character*. The rules of correspondence between input and output data for modules within the MTT fully preserve information equivalence at all language levels.
- *Variety of structures and formalisms*. Each module has its own rules and formalisms in the MTT, because of significant variety of structures reflecting data on different levels (strings, trees, and networks, correspondingly). On each level, the MTT considers just a minimal possible set of descriptive features. On the contrary, the generative grammar tradition tries to find some common formalism covering the whole language, so that the total

multiplicity of features of various levels are considered jointly, without explicit division to different levels.

- *Peculiarities in deep and surface syntactic*. The entities and syntactic features of these two levels are distinctly different in the MTT. Auxiliary and functional words of a surface disappear at the depth. Analogously, some syntactic characteristics of wordforms are present only at the surface (e.g., agreement features of gender and number for Spanish adjectives), whereas other features, being implied by meaning, are retained on the deeper levels as well (e.g., number for nouns). Such separation facilitates the minimization of descriptive means on each level. The notions of deep and surface syntactic levels in Chomskian theory too, but as we could already see, they are defined there in a quite different way.

- *Independence between the syntactic hierarchy of words and their order in a sentence*. These two aspects of a sentence, the labeled dependency trees and the word order, are supposed to be implied by different, though interconnected, factors. Formally, this leads to the systematic use of dependency grammars on the syntactic level, rather than of constituency grammars. Therefore, the basic rules of inter-level transformations turned out to be quite different in the MTT, as compared to the generative grammar. The basic advantage of dependency grammars is seen in that the links between meaningful words are retained on the semantic level, whereas for constituency grammars (with the exception of HPSG) the semantic links have to be discovered by a separate mechanism.

- *Orientation to languages of a type different from English*. To a certain extent, the opposition between dependency and constituency grammars is connected with different types of languages. Dependency grammars are especially appropriate for languages with free word order like Latin, Russian or Spanish, while constituency grammars suit for languages with strict word order as English. However, the MTT is suited to describe such languages as English, French, and German too. Vast experience in opera-

tions with dependency trees is accumulated in frame of the MTT, for several languages. The generative tradition (e.g., HPSG) moves to the dependency trees too, but with some reservations and in some indirect manner.

- *Means of lexical functions and synonymous variations*. Just the MTT has mentioned that the great part of word combinations known in any language is produced according to their mutual lexical constraints. For example, we can say in English *heart attack* and *cordial greetings*, but neither *cardiac attack* nor *hearty greeting*, though the meaning of the lexemes to be combined permit all these combinations. Such limitations in the combinability have formed the calculus of the so-called lexical functions within the MTT. The calculus includes rules of transformation of syntactic trees containing lexical functions from one form to another. A human can convey the same meaning in many possible ways. For example, the Spanish sentence *Juan me prestó ayuda* is equal to *Juan me ayudó.* Lexical functions permit to make these conversions quite formally, thus implementing the mechanism of synonymous variations. This property plays the essential role in synthesis and has no analog in the generative tradition. When translating from one language to another, a variant realizable for a specific construction is searched in the target language among synonymous syntactic variants. Lexical functions permit to standardize semantic representation as well, diminishing the variety of labels for semantic nodes.

- *Government pattern*. In contradistinction to subcategorization frames of generative linguistics, government patterns in the MTT directly connect semantic and syntactic valencies of words. Not only verbs, but also other parts of speech are described in terms of government patterns. Hence, they permit to explicitly indicate how each semantic valency can be represented on the syntactic level: by a noun only, by the given preposition and a noun, by any of the given prepositions and a noun, by an infinitive, or by any other way. The word order is not fixed in government pat-

terns. To the contrary, the subcategorization frames for verbs are usually reduced just to a list of all possible combinations of syntactic valencies, separately for each possible order in a sentence. In languages with rather free word order, the number of such frames for specific verbs can reach a few dozens, and this obscures the whole picture of semantic valencies. Additionally, the variety of sets of verbs with the same combination of subcategorization frames can be quite comparable with the total number of verbs in such languages as Spanish, French or Russian.

- *Keeping traditions and terminology of classical linguistics*. The MTT treats the heritage of classical linguistics much more carefully than generative computational linguistics. In its lasting development, the MTT has shown that even the increased accuracy of description and the necessity of rigorous formalisms usually permits to preserve the existing terminology, perhaps after giving more strict definitions to the terms. The notions of *phoneme, morpheme, morph, grammeme, lexeme, part of speech, agreement, number, gender, tense, person, syntactic subject, syntactic object, syntactic predicate, actant, circonstant*, etc., have been retained. In the frameworks of generative linguistics, the theories are sometimes constructed nearly from zero, without attempts to interpret relevant phenomena in terms already known in general linguistics. These theories sometimes ignored the notions and methods of classical linguistics, including those of structuralism. This does not always give an additional strictness. More often, this leads to terminological confusion, since specialists in the adjacent fields merely do not understand each other.

REDUCED MODELS

We can formulate the problem of selecting a good model for any specific linguistic application as follows.

A *holistic* model of the language facilitates describing the language as a whole system. However, when we concentrate on the

objectives of a specific application system, we can select for our purposes only that level, or those levels, of the whole language description, which are relevant and sufficient for the specific objective. Thus, we can use a *reduced* model for algorithmization of a specific application.

Here are some examples of the adequate choice of such a reduced description.

- If we want to build an information retrieval system based on the use of keywords that differ from each other only by their invariant parts remaining after cutting off irrelevant suffixes and endings, then no linguistic levels are necessary. All words like *México*, *mexicanos*, *mexicana*, etc., can be equivalent for such a system. Other relevant groups can be *gobierno*, *gobiernos*, or *ciudad*, *ciudades*, etc. Thus, we can use a list containing only the initial substrings (i.e., stems or quasi-stems) like *mexic-*, *gobierno-*, *ciudad-*, etc. We also will instruct the program to ignore the case of letters. Our tasks can be solved by a simple search for these substrings in the text. Thus, linguistic knowledge is reduced here to the list of substrings mentioned above.

- If we want to consider in our system the wordforms *dormí*, *duermo*, *durmió*, etc., or *será*, *es*, *fui*, *era*, *sido*, etc. as equivalent keywords, then we must introduce the morphologic level of description. This gives us a method of how to automatically reduce all these wordforms to standard forms like *dormir* or *ser*.

- If we want to distinguish in our texts those occurrences of the string *México* that refer to the name of the city, from the occurrences that refer to name of the state or country, then we should introduce both morphologic and syntactic levels. Indeed, only word combinations or the broader contexts of the relevant words can help us to disambiguate such word occurrences.

- In a spell checker without limitations on available memory, we can store all wordforms in the computer dictionary. Nevertheless, if the memory is limited and the language is highly inflectional,

like Spanish, French or Russian, we will have to use some morphologic representation (splitting words to stems and endings) for all the relevant wordforms.

- In grammar checkers, we should take morphologic and syntactic levels, in order to check the syntactic structures of all the sentences. The semantic level usually remains unnecessary.

- For translation from one natural language to another, rather distant, language, all the linguistic levels are necessary. However, for translation between two very similar languages, only morphologic and syntactic levels may be necessary. For the case of such very "isomorphic" languages as Spanish and Portuguese, the morphologic level alone may suffice.

- If we create a very simple system of understanding of sentences with a narrow subject area, a small dictionary, and a very strict order of words, we can reduce the dictionary to the set of strings reflecting initial parts of the words actually used in such texts and directly supply them with the semantic interpretations. In this way, we entirely avoid the morphologic and syntactic problems; only the textual and the semantic levels of representation are necessary.

- If we create a more robust system of text understanding, then we should take a full model of language plus a reasoning subsystem, for the complete semantic interpretation of the text.

However, to make a reasonable choice of any practical situation, we need to know the whole model.

DO WE REALLY NEED LINGUISTIC MODELS?

Now let us reason a little bit on whether computer scientists really need a generalizing (complete) model of language.

In modern theoretical linguistics, certain researchers study phonology, the other ones morphology, the third ones syntax, and the

fourth ones semantics and pragmatics. Within phonology, somebody became absorbed in accentuation, within semantics, in speech acts, etc. There is no limit to the subdivision of the great linguistic science, as well as there is seemingly no necessity to occupy oneself once more, after ancient Greeks, Ferdinand de Saussure and Noam Chomsky, with the philosophical question "What is natural language and what should its general model be?"

The main criteria of truth in theoretical linguistic research are its logical character, consistency, and correspondence between intuitive conceptions about the given linguistic phenomena of the theory's author and of other members of linguists' community.

In this sense, the works of modern specialists in theoretical linguistics seem to be just stages of inner development of this science. It often seems unnecessary to classify them according to whether they support or correspond to any complete model.

The situation in computational linguistics is somewhat different. Here the criterion of truth is the proximity of results of functioning of a program for processing language utterances to the ideal performance determined by mental abilities of an average speaker of the language. Since the processing procedure, because of its complexity, should be split into several stages, a complete model is quite necessary to recommend what formal features and structures are to be assigned to the utterances and to the language as a whole on each stage, and how these features should interact and participate at each stage of linguistic transformations within computer. Thus, all theoretical premises and results should be given here quite explicitly and should correspond to each other in theirs structures and interfaces.

Theoreticians tell us about the rise of experimental linguistics on this basis. It seems that in the future, experimental tests of the deepest results in all "partial" linguistic theories will be an inevitable element of evolution of this science as a whole. As to computational linguistics, the computerized experimentation is crucial right now, and it is directly influenced by what structures are selected for lan-

guage description and what processing steps are recommended by the theory.

Therefore, the seemimgly philosophical problem of linguistic modeling turned out to be primordial for computational linguistics. Two linguistic models selected from their vast variety will be studied in this book in more detail.

ANALOGY IN NATURAL LANGUAGES

Analogy is the prevalence of a pattern (i.e., one rule or a small set of rules) in the formal description of some linguistic phenomena. In the simplest case, the pattern can be represented with the partially filled table like the one on the page 20:

revolución	*revolution*
investigación	?

The history of any natural language contains numerous cases when a phonologic or morphologic pattern became prevailing and, by analogy, has adduced words with similar properties.

An example of analogy in Spanish phonology is the availability of the *e* before the consonant combinations *sp-, st-*, *sn-*, or *sf-* at the beginning of words. In Latin, the combinations *sp-* and *st-* at the initial position were quite habitual: *specialis, spectaculum, spiritus, statua, statura,* etc.

When Spanish language was developed from Vulgar Latin, all such words had been considered uneasy in their pronunciation and have been supplied with *e-*: *especial, espectáculo, espíritu, estatua, estatura,* etc. Thus, a law of "hispanicizing by analogy" was formed, according to which all words with such a phonetic peculiarity, while loaned from any foreign language, acquire the *e* as the initial letter.

We can compile the following table of analogy, where the right column gives Spanish words after their loaning from various languages:

statura (Lat.)	*estatura*
sphaira (Gr.)	*esfera*
slogan (Eng.)	*eslogan*
smoking (Eng.)	*esmoquin*
standardize (Eng.)	*estandarizar*

As another example, one can observe a multiplicity of nouns ending in -*ción* in Spanish, though there exist another suffixes for the same meaning of action and/or its result: -*miento*, -*aje*, -*azgo*, -*anza*, etc. Development of Spanish in the recent centuries has produced a great number of -*ción*-words derived by analogy, so that sometimes a special effort is necessary to avoid their clustering in one sentence for better style. Such a stylistic problem has been even called *cacophony*.

Nevertheless, an important feature of language restricts the law of analogy. If the analogy generates too many homonyms, easy understanding of speech is hampered. In such situations, analogy is not usually applied.

A more general tendency can be also observed. Lexicon and levels of natural language are conceptual systems of intricately interrelated subsystems. If a feature of some subsystem has the tendency to change and this hinders the correct functioning of another subsystem, then two possible ways for bypassing the trouble can be observed. First, the innovation of the initiating subsystem can be not accepted. Second, the influenced subsystem can also change its rules, introducing in turn its own innovations.

For example, if a metonymic change of meaning gives a new word, and the new word frequently occurs in the same contexts as the original one, then this can hinder the comprehension. Hence, either the novel or the original word should be eliminated from language.

In modern languages, one can see the immediate impact of analogy in the fact that the great amount of scientific, technical, and political terms is created according to quite a few morphologic rules. For example, the Spanish verbs *automatizar, pasteurizar,*

globalizar, etc., are constructed coming from a noun (maybe proper name) expressing a conception (*autómata, Pasteur, globo*, etc.) and the suffix -*izar*/-*alizar* expressing the idea of subjection to a conception or functioning according to it.

Computational linguistics directly uses the laws of analogy in the processing of unknown words. Any online dictionary is limited in its size so that many words already known in the language are absent in it (say, because these words appear in the language after the dictionary was compiled). To "understand" such words in some way, the program can presuppose the most common and frequent properties.

Let us imagine, for instance, a Spanish-speaking reader who meets the word *internetizarán* in a text. Basing on the morphologic rules, he or she readily reconstructs the infinitive of the hypothetical verb *internetizar*. However, this verb is not familiar either, whereas the word *Internet* could be already included in his or her mental dictionary. According to the analogy implied by -*izar*, the reader thus can conclude that *internetizar* means 'to make something to function on the principles of Internet.'

A natural language processor can reason just in the same way. Moreover, when such a program meets a word like *linuxizar* it can suppose that there exists a conception *linux* even if it is absent in the machine dictionary. Such supposition can suggest a very rough "comprehension" of the unknown word: 'to make something to function on the principles of *linux*,' even if the word *linux* is left incomprehensible.

EMPIRICAL VERSUS RATIONALIST APPROACHES

In the recent years, the interest to *empirical* approach in linguistic research has livened. The empirical approach is based on numerous statistical observations gathered purely automatically. Hence, it can be called *statistical* approach as well. It is opposed to the *rationalist* approach, which requires constructing a functional model of language on the base of texts and the researcher's intuition. Through-

out this book, we explain only the rationalist approach, both in the variants of the generative grammar, and of the MTT.

The empirical approach can be illustrated more easily on the example of the machine translation. A huge bilingual corpus of text is being taken, i.e., two very long, equal in the meaning, and arranged in parallel, texts in two different languages. Statistics is being gathered on text fragments going in nearly equal places on the opposite sides of the bilingual. An attempt is being made to learn how, for any fragment in one language (including those not yet met in the corpus), to find a fragment in other language, which is equivalent to the former in the meaning. The solution would give the translation of any text by the empirical method.

It can be seen that such a method unites two types of models given above—research and functional ones. It is also obvious that it is impossible to accumulate the statistics in general, without elaboration of some definitions and specifications.

It is first necessary to determine what is the size of fragments to be compared, what are "nearly equal" places and what is the equivalence (or rather quasi-equivalence) of the fragments in the two parallel texts. Hence, the answer to these questions requires some elements of a rationalist model, as it was exposed everywhere above.

It is difficult to deny the need of statistical observations in computational linguistics. In particular, in any rationalist model we should take into account those linguistic phenomena (lexemes, syntactic constructions, etc.) that can be met in texts most frequently. That is why we will spare the space in this book for statistical methods in computational linguistics.

As to the empirical method just outlined, its real successes are still unknown.

A common feature of rationalist and empiric methods is that both of them presuppose natural language cognizable and algorithmizable. Linguists and philosophers suggest sometimes the opposite point of view. They argue that since human beings usually reason without any limitations of logic, their language activity can also lack a logical and algorithmic basis.

As applied to the natural language, this pessimistic viewpoint however contradicts to the everyday human practice, as well as to the practice of modern computational linguistics. Humans can manage a new natural language in any age and in a rather rational manner, whereas computational linguistics has already managed some features of natural language, and the process of mastering is going on. Thus, we may neglect this pessimistic point of view.

LIMITED SCOPE OF THE MODERN LINGUISTIC THEORIES

Even the most advanced linguistic theories cannot pretend to cover all computational problems, at least at present. Indeed, all of them evidently have the following limitations:

- Only the problems of morphology and syntax are under intensive elaboration in these theories, whereas semantics is investigated to a significantly lesser degree. The goal of atomization and quite consistent decomposition of semantic components remained unattained. The more limited problem of rational decomposition of word meaning, i.e., of the semantic representation for a given lexeme through the meaning of some other more simple ones, is not yet solved on a large scale in any language. This is the great problem of lexical semantics. It develops well, but computational linguistics considers this development too slow and lacking immediate connection with computations.
- Modern semantics cannot yet formalize its problems adjacent to pragmatics to the degree sufficient for applications. Indeed, there is no complete theory of links between the meaning of text and the goals of this text in a practical situation, as well as between the speaker's intentions and the listener's perception, though there are numerous valuable observations on these aspects. For example, computational linguistics cannot distinguish that the Spanish utterance *¿Dónde está la sal?* is a real query for information about the salt, whereas *¿Podría usted pasarme la sal?* is a request for the specific action given in a polite form. As another

example, no automate can "comprehend" so far that the sentence *Niños son niños* is not a trivial tautology, but the idea that children have specific features of their own and thus should be treated properly. To cope with such intricacies, linguists should model the human world with respect to its customs, habits, etiquette, relations between generations, etc. This is an *extralinguistic* knowledge of *encyclopedic* type. Until now, computational linguistics and artificial intelligence do not know how to effectively distinguish, to assemble apart and then to combine the knowledge of purely linguistic and evidently encyclopedic type. What is more, a "dictionary," which specifies all encyclopedic information needed for comprehension of texts rather simple for a human to understand, would be so huge that it is unlikely to be compiled with sufficient completeness in the nearest future.

- The results of the recent investigations mainly cover separate sentences, but not discourses. The complicated semantics of discourse, including the information on referential links between different entities, a target matter under explanation, current author's estimations, and the author's planning of the discourse, still waits its deeper research.

- It is well known in theoretical linguistics, that the set of wordforms comprising a sentence is chosen according to the main matter of this sentence, whereas the word order depends both on this wordform set (e.g., a preposition should precede the related noun) and on *communicative structure of a text*. This notion reflects what the author considers already known or presupposed at this stage of the communication process (i.e. topic) and what information he or she chooses to communicate right now (i.e. comment). In generative grammars, the variations of word order depending on communicative structure of texts had not been even noticed for a long time. The MTT and general linguistics as a whole give now a rather elaborated informal study of these problems. For example, this study explains the obvious difference in meaning between Spanish sentences *Juan llegó* 'Juan came' and

Llegó Juan 'It is Juan who came,' where the same words go in different order. As a more complicated example using the same wordforms but in different order, the sentence *En México se habla el español* 'In Mexico, Spanish is spoken' turns to be unconditionally true, while the meaning of *El español se habla en México* 'Spanish is spoken in Mexico' is quite different and dubious, since Spanish is spoken not only in Mexico. In spite of all the theoretical advances, the global formalization of communicative structures is not yet attained. So far, these advances cannot be used for either text synthesis or analysis.

- The problem of how people learn natural language in their childhood remains unsolved. The idea of linguistic universalities once introduced in general linguistics has transformed now into the idea of the *Universal Grammar* by Chomsky. All languages are considered species of this grammar, with a finite set of generalized features supposedly adjustable to a specific "option" (Spanish, English, etc.). Newborn children are supposed to have the Universal Grammar in their brains, and their adaptation to a specific language is accomplished at childhood. However, the goal to discover the structure and laws of the Universal Grammar remains unattained until now. Thus, computational linguistics cannot propose any universal algorithms equally applicable to various languages.

Even proponents of the contemporary linguistic theories do not believe that all facts of languages can be interpreted through their favorite ideas, to solve current problems of computational linguistics. Meanwhile, the science advances, maybe slower than we wish.

The readers of this book will be able to learn from it a multiplicity of already known linguistic facts and laws. In the same time, they can realize that numerous interesting and very difficult problems, with which computational linguistics is faced nowadays, stay yet unsolved. They are still awaiting a Chomsky of their own.

Conclusions

A linguistic model is a system of data (features, types, structures, levels, etc.) and rules, which, taken together, can exhibit a "behavior" similar to that of the human brain in understanding and producing speech and texts. A functional linguistic model takes into account the observed language behavior of human beings rather than the physiological activity of the brain. This behavior is reflected in the texts or speech they produce in response to the texts or speech they perceive.

So far, the direct modeling of the brain structures has failed, and several functional models were proposed for the sake of computational linguistics. The modern functional models have many features in common. They are intended to be quite formal, have a dynamic and non-generative character, provide independence of linguistic algorithms from linguistic data, and consider dictionaries as one of the main, inalienable parts of the model.

Theoretical approaches provide a solid basis for both holistic and reduced models of language oriented to applications. The degree of the reduction in such a model heavily depends on the specific application.

EXERCISES

THIS SECTION CONTAINS some review questions recommended to the readers to verify their correct understanding of the contents of the book, and the problems recommended for exams.

REVIEW QUESTIONS

THE FOLLOWING QUESTIONS can be used to check whether the reader has understood and remembered the main contents of the book. The questions are also recommended for the exam on this course of Computational Linguistics The questions marked with the sign ° are the most important ones.

1. Why is automatic processing of natural language important for the humankind?
2. Why are theoretical aspects of linguistics necessary for computational linguistics?
3. How are related the methods of computational linguistics and of artificial intelligence?
4. How is coordinated computational linguistics with computer science?
5. What is general linguistics?
6. What aspects of natural language do phonology, morphology, syntax, semantic, and pragmatic study?
7. What is historical linguistics? Contrastive linguistics? Sociolinguistics?
8. What is dialectology? What is Mexican Spanish with respect to Spanish language?
9. What is lexicography? Why is it important for NL process-

ing?

10. What are the narrower and the broader comprehension of mathematical linguistics?
11. What is computational linguistics? How is it related with applied linguistics?
12. ° What is the structuralist approach in general linguistics?
13. What are constituents? What is constituency tree?
14. What mathematical means were proposed by Noam Chomsky? What purposes can they serve for?
15. What example of context-free grammar to generate simple sentences do you know?
16. What are transformation grammars?
17. What are valencies in linguistics? What is the difference between syntactic and semantic valencies?
18. What are subcategorization frames and how they describe valencies of verbs?
19. What are constraints in computational linguistics?
20. What is the notion of head in Head-driven Phrase Structure Grammar?
21. What is the idea of unification in computational linguistics?
22. Why should language be viewed as a transformer?
23. Why should this transformer be considered to contain several stages of transformation?
24. What is meant by Text in the Meaning ⇔ Text Theory?
25. What is meant by Meaning in the Meaning ⇔ Text Theory?
26. What are the main levels of language representation? Which levels are called surface ones, and which deep one? Why?
27. What are dependency tree in computational linguistics?
28. What are the two methods of information representation on semantic level? What are the semantic labels? Are they

words?

29. What are government patterns in the Meaning ⇔ Text Theory? How they describe syntactic and semantic valencies?
30. What are the main applications and classes of applications of computational linguistics?
31. What linguistic knowledge is used in hyphenation programs? Spell checkers? Grammar checkers? Style checkers?
32. What linguistic knowledge is used in information retrieval systems? In what a way does this knowledge influence the main characteristics of information retrieval systems?
33. How can we determine automatically the theme of the document?
34. How is the linguistic knowledge used in automatic translation? What are the main stages of automatic translation? Are all of these stages always necessary?
35. What is automatic text generation?
36. What are specifics of natural language interfaces?
37. What is extraction of factual data from texts?
38. What is language understanding? What linguistic knowledge should it employ? What are the main difficulties in creation of systems for language understanding?
39. What is EuroWordNet?
40. Do optical character recognition and speech recognition require linguistic knowledge?
41. What is modeling in general?
42. What kinds of linguistic modeling do you know? What are research linguistic models used for?
43. What are functional models in linguistics? What are their common features?
44. Are the Meaning ⇔ Text Theory and Head-driven Phrase

Structure Grammar functional models?

45. What are specific features of the Meaning ⇔ Text model?
46. What are holistic and reduced models? Is the most detailed and broad model always the better one?
47. What aspects of language are not covered by modern linguistic models?
48. ◦ *Word*, *wordform*, and *lexeme*, what is the difference between them? When can we use each of them?
49. What is synonymy? What kinds of synonyms exist? Can synonymy be avoided in natural language?
50. What is homonymy? What kinds of homonyms exist? Can homonymy be avoided in natural language?
51. What are metaphoric and metonymic methods of creation of new words in natural language?
52. What are the specifics of computer-based dictionaries?
53. What is analogy in linguistics? How can we use it for NL processing?
54. What is empirical approach in linguistics? To what kind of problems can it be applied?
55. ◦ What is a sign? What is a linguistic sign?
56. What is the syntactics of a linguistic sign in the Meaning ⇔ Text Theory?
57. What is the structure of the linguistic sign in Head-driven Phrase Structure Grammar?
58. Are there any commonalities in the linguistic description between generative, Meaning ⇔ Text, and constraint-based approaches?

PROBLEMS RECOMMENDED FOR EXAMS

IN THIS SECTION, each test question is supplied with a set of four variants of the answer, of which exactly one is correct and the others are not.

1. Why automatic natural language processing (NPL) is important for the humankind?

   A. Because NPL takes decisions in place of humans.
   B. Because NPL facilitates humans to prepare, to read, and to search through many texts.
   C. Because NPL permits humans not to read any texts by themselves.
   D. Because NPL facilitates humans to use computers.

2. Why theoretical aspects of linguistics are necessary for computational linguistics?

   A. Because they help to prepare good user's manuals for products of computational linguistics.
   B. Because they help to evaluate the performance of computational linguistics products.
   C. Because they help to gather statistics of various language elements.
   D. Because they help to comprehend general structure of languages.

3. How does computational linguistics (CL) coordinate with artificial intelligence (AI)?

   A. CL is a part of AI.
   B. AI is a part of CL.
   C. CL does not coordinate with AI at all.
   D. CL and AI have many tools in common.

4. How does computational linguistics (CL) coordinate with computer science (CS)?

   A. CL is a part of CS.
   B. CS is a part of CL.

C. CS is a tool for CL. D. CL is a tool for CS.

5. What does general linguistics study?

   A. Laws of orthography.
   B. Laws and structures of languages.
   C. Rules of good word usage.
   D. Rules of good style.

6. What does phonology study?

   A. Sounds of music.
   B. Sounds uttered by animals.
   C. Sounds forming words for their distinguishing.
   D. Sounds of alarm.

7. What does morphology study?

   A. How to combine words to sentences.
   B. How to combine sounds or letters to words.
   C. How to form abbreviations.
   D. How to form composed words like *rascacielos*.

8. What does syntax study?

   A. How to combine parts of words to words.
   B. How to compose a text of paragraphs.
   C. How to compose a paragraph of sentences.
   D. How to combine words to phrases and sentences.

9. What does semantics study?

   A. How humans think.
   B. How humans code the meaning in their brains.
   C. How humans perceive the outer world.
   D. How human express their wishes.

10. What does historical linguistics study?

   A. Biographies of eminent linguists.
   B. Theories of language origin in the prehistoric times.
   C. Evolution of different languages in the course of time.

D. History of development of grammars.

11. What does contrastive linguistics study?

A. Controversies between different linguistic theories.
B. Differences between various languages.
C. Antonyms like *pequeño–grande* ‘small–big’.
D. Similarities of various non-cognate languages in their structures.

12. What part of linguistics studies peculiarities of Spanish in the Yucatan peninsula?

A. Historical linguistics.
B. Dialectology.
C. Sociolinguistics.
D. Comparative linguistics.

13. What does lexicography study?

A. Rules of orthography.
B. Rules of good word usage.
C. Pronunciation of words.
D. Description of all words in languages.

14. What does computational linguistics study?

A. How to count words and other linguistic elements in texts.
B. How to create programs for automatic text processing.
C. How to teach a foreign language with the aid of a computer.
D. How to prepare correct text with the aid of a computer.

15. How is computational linguistics (CL) related with applied linguistics (AL)?

A. CL is a part of AL.
B. AL is a part of CL.
C. AL is equal to CL.
D. AL and CL are independent branches of linguistics.

16. What are constituents in linguistic description?

A. Arbitrary parts of a sentence.
B. Contiguous groups of words in a sentence.

C. Words with the most important meaning in a sentence.
D. All words in a sentence except for auxiliary ones.

17. What does a constituency tree contain as its nodes?

A. Various words.
B. Various grammatical categories.
C. Various sentences.
D. Various word endings.

18. What mathematical means did Chomsky propose?

A. Hierarchy of generative grammars.
B. Algorithms of language analysis.
C. Normative grammars for several languages.
D. Modified version of English grammar.

19. What can transformation grammars describe?

A. How to shorten context-free sentences.
B. How to repeat context-free sentences.
C. How to transit from a context-free sentence to its negative or interrogative version.
D. How to generate context-free sentences.

20. What is valency in linguistics?

A. A label at a word.
B. A link from one word to another.
C. A prepositional phrase.
D. A part of a labeled tree.

21. What is the difference between syntactic and semantic valencies?

A. Syntactic valencies link some words into pairs, while semantic valencies link other pairs.
B. Syntactic and semantic valencies link the same pairs of words, but in opposite directions.
C. Syntactic valencies describe the links between the same words as semantic valencies, but on different levels.

D. Syntactic and semantic valencies are essentially the same.

22. What is the notion of *head* in Head-driven Phrase Structure Grammar?

A. The principal subconstituent.
B. The center subconstituent.
C. The leftmost subconstituent.
D. The constituent that covers all its subconstituents.

23. What is unification in computational linguistics?

A. Standardization of features of wordforms.
B. Reducing wordforms to their dictionary forms.
C. Revealing similarities of features of different wordforms and uniting feature sets.
D. Uniting structures of several sentences into a common structure.

24. What is dependency tree in computational linguistics?

A. The same as constituency tree.
B. A labeled hierarchy of immediate links between wordforms in a sentence.
C. Hierarchy of meanings represented by words in a sentence.
D. Hierarchy of features assigned to wordforms in a sentence.

25. What applications of computational linguistics are the most developed now?

A. Grammar checking.
B. Spell checking.
C. Style checking.
D. Language understanding.

26. What applications of computational linguistics are the least developed now?

A. Grammar checking.
B. Language understanding.
C. Style checking.
D. Information retrieval.

27. What linguistic knowledge is used for automatic hyphenation?

A. How to use various fonts for different words.
B. What letters are vowels and consonants.

C. How to use lowercase and uppercase letters in writing.
D. How to combine usual words and numbers in a text.

28. What linguistic knowledge is used for spell checking?

A. How to use lowercase and uppercase letters in writing.
B. What are the laws of morphologic variations for words in this language.
C. What are rules of hyphenation in this language.
D. What words can stay adjacent in a text.

29. What linguistic knowledge is sufficient for grammar checking?

A. What syntactical constructions are correct in this language.
B. What words are supposedly correct in this language.
C. What phrases are commonly used in this language.
D. What words can stay adjacent in a text.

30. What linguistic knowledge is used for style checking?

A. What punctuation marks are appropriate in such a context.
B. How to arrange parts of a text in a proper sequence.
C. What words are more appropriate in a context.
D. How to split a text to adequate parts.

31. What linguistic knowledge is used in information retrieval systems?

A. Inverse files of terms.
B. Dictionaries and thesauri, consisting of terms.
C. Quick search algorithms.
D. Keyword sets at each document.

32. What are the main characteristics of information retrieval systems?

A. Response time.
B. Recall and precision.
C. Necessary size of memory for delivery.
D. User-friendliness of the interface.

33. How can we better determine automatically the theme of the document?

    A. By considering the “hottest” themes for the present moment.
    B. By considering the origin of the document.
    C. By considering the words, which the document uses.
    D. By considering the addressee of the document.

34. What is automatic text generation?

    A. Deriving a text from some formal specifications.
    B. Selecting entries in a preliminary prepared list of phrases.
    C. Generating phrases basing on statistics of various language elements.
    D. Derivation process of some generative grammar.

35. What is extraction of factual data from a text?

    A. Determining what is the time of creation and the size of the text file.
    B. Determining what audience this text is oriented to.
    C. Determining qualitative and quantitative features of events, persons or things, which are touched upon in the text.
    D. Determining linguistic means used in the text.

36. What is language understanding by a computer?

    A. Transforming text into a binary code.
    B. Transforming text into a graph representation.
    C. Transforming text into a form that conserves the meaning and is directly usable by purposeful automata.
    D. Transforming text into a table representation.

37. What are the main difficulties in creation of systems for language understanding?

    A. Modern computers are insufficiently quick to solve the problem.
    B. The ways of coding of meaning in texts are very complicated and are not sufficiently investigated.

C. Modern computers have insufficiently big memory to solve the problem.
D. Syntactic analysis gives too many variants.

38. What is WordNet (EuroWordNet)?

A. A usual dictionary, but in electronic form.
B. A thesaurus with a developed network of semantic links.
C. An electronic dictionary of synonyms.
D. An electronic dictionary in which we can find the part—whole links between words.

39. What linguistic knowledge does optical character recognition require?

A. How to use lowercase and uppercase letters in writing.
B. What strings of letters are correct words in writing.
C. What are rules of hyphenation in this language.
D. What words can stay adjacent in a text.

40. What linguistic knowledge does speech recognition require?

A. What variations of intonation do exist in this language.
B. What variations of logical stress do exist in this language.
C. What sequences of sounds are correct words in speech.
D. What words can stay adjacent in a speech in this language.

41. What is natural language?

A. Principal means for expressing human thoughts.
B. Principle means for text generation.
C. Bi-directional transformer Meaning ⇔ Text.
D. Principal means of human communication.

42. What is a model in general?

A. It is an important part of the modeled system.
B. It imitates the most important features of the modeled system.
C. It includes the modeled system as the most important part.

D. It is connected with the modeled system within a system of higher rank.

43. What is the reduced model of a language?

A. It reflects all linguistic levels, but to different degree.
B. It models linguistic levels most important for the applied system.
C. It models surface linguistic levels.
D. It models deep linguistic levels.

44. What aspect of language is the least explored by modern linguistics?

A. Morphology.
B. Syntax.
C. Phonology.
D. Semantics.

45. What is a lexeme?

A. A set of letters.
B. A string of letters.
C. A set of wordforms with the same meaning.
D. A common meaning of several wordforms.

46. What entity forms the entry in a common vocabulary?

A. A word.
B. A wordform.
C. A lexeme.
D. A morph.

47. How many word occurrences are in the sentence *Yo te amo, pero tú no me contesta como yo* 'I love you but you do not return me my love'?

A. Twelve.
B. Ten.
C. Nine.
D. Seven.

48. How many wordforms are in the sentence *Yo te amo, pero tú no me contesta como yo* 'I love you but you do not return me my love'?

A. Twelve.
B. Ten.
C. Nine.
D. Seven.

49. How many lexemes are there in the sentence *Yo te amo, pero tú no me contestas como yo* ‘I love you but you do not return me my love’?

A. Twelve.
B. Ten.
C. Nine.
D. Seven.

50. What pair of the following ones consists of synonyms?

A. *escoger*, *optar* ‘choose, opt’.
B. *tener, obtener* ‘have, obtain’.
C. *fuerza, debilidad* ‘power, weakness’.
D. *árbol, manzana* ‘tree, apple’.

51. What are synonyms?

A. The signifieds are different, but the signifiers are equal.
B. The signifiers are different, but the signifieds are equal.
C. The signifieds are different, and the signifiers are different.
D. The signifieds are equal, and the signifiers are equal.

52. What are homonyms?

A. The signifieds are different, but the signifiers are equal.
B. The signifiers are different, but the signifieds are equal.
C. The signifieds are different, and the signifiers are different.
D. The signifieds are equal, and the signifiers are equal.

53. By what method used in order to enrich natural languages the Spanish words *escuela* and *teatro* have acquired the meaning ‘corresponding building’?

A. By metaphor.
B. By metonymy.
C. By loaning from other language.
D. By assigning a new meaning to an old word at random.

54. How many components does a linguistic sign have?

A. One
B. Two
C. Three
D. More than three.

LITERATURE

IN THIS SECTION we list recommended and additional literature for more profound study of the topic of this book, as well as bibliographic references. The reader is also encouraged to study other publications of the authors of this book, which can be found on www.Gelbukh.com.

RECOMMENDED LITERATURE

1. Allen, J. *Natural Language Understanding*. The Benjamin / Cummings Publ., Amsterdam, Bonn, Sidney, Singapore, Tokyo, Madrid, 1995.
2. Cortés García, U., J. Béjar Alonso, A. Moreno Ribas. *Inteligencia Artificial*. Edicions UPC, Barcelona, 1993.
3. Grishman, R. *Computational linguistics. An introduction.* Cambridge University Press, 1986.
4. Jurafsky, D., J. H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice-Hall, 2000; see www.cs.colorado.edu/~martin/slp.html.
5. Manning, C., H. Schütze, Foundations of Statistical Natural Language Processing, MIT Press, 1999; www-nlp.stanford.edu/fsnlp.
6. Mel'čuk, I. A. *Dependency Syntax: Theory and Practice.* State University of New York Press, NY, 1988.
7. Pollard, C., and I. A. Sag. *Head-driven Phrase Structure grammar.* CSLI Publ., Stanford. University of Chicago Press, Chicago and London, 1994.

8. Sag, I. A., and T. Wasow. *Syntactic theory*: *Formal Introduction.* CSLI Publ., Stanford University of Chicago Press, Chicago and London, 1999; see also http://hpsg.stanford.edu/hpsg.
9. Sell, P. *Lectures on Contemporary Syntactic Theories*. CSLI Publ., Stanford, 1985.

ADDITIONAL LITERATURE

10. Baeza-Yates, R., B. Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley Longman and ACM Press, 1999.
11. Beristáin, Helena. *Gramática estructural de la lengua española*. UNAM. Limusa Noriega Editores. México, 1996.
12. Chomsky, N. *Syntactic Structures*. The Hague: Mouton, 1957.
13. Fillmore, C. J. *The case for case*. In: Bach, E., and B.T. Halms (eds.), Universals in linguistic theory. NY, Chicago, San Francisco, 1968
14. Fuentes, J. L. *Gramática moderna de la lengua española.* Biográfica Internacional, Colombia, 1988.
15. Gartz, Irene. *Análisis de las estructuras del español.* Editorial Trillas, México, 1991.
16. Leroy, M. *Las grandes corrientes de la lingüística*, Fondo de Cultura Económica, 1982.
17. Mel'čuk, I. A. *Dependency in Linguistic Description*. Benjamin Publ. Company, 1999.
18. Saussure, Ferdinand de. *Curso de lingüística general.* Editorial Fontamara, México, 1996.
19. Steele, J. (ed). *Meaning–Text Theory. Linguistics, Lexicography, and Implications.* University of Ottawa Press, Ottawa, 1990.

GENERAL GRAMMARS AND DICTIONARIES

20. Criado de Val, M. *Gramática española*. Madrid, 1958.
21. Cuervo, R. J. *Diccionario de construcción y régimen de la lengua castellana*. Instituto de Cara y Cuervo. Bogotá, 1953.
22. *Diccionario De La Lengua*. Grupo Anaya. http://www.anaya.es/diccionario/diccionar.htm.
23. *Diccionario de la Real Academia Española*. Real Academia Española. Edición en CD-ROM, Espasa-Calpe, Madrid.
24. *Diccionario Océano de sinónimos y antónimos.* Océano Grupo Editorial, Barcelona, 1997.
25. Gilli Gaya, S. *Curso superior de sintaxis española*. Barcelona, 1964.
26. Lara Ramos, L. F. (dir.). *Diccionario del Español Usual en México*. El Colegio de México, 1996.
27. Martínez Amador, E. *Diccionario gramatical y de dudas del idioma*. Barcelona, 1953.
28. Moliner, María. *Diccionario de uso del español.* Edición CD-ROM. Editorial Gredos, 1998.
29. Moneva Puyol, J. M. *Gramática castellana*. Barcelona—Madrid—Buenos Aires—Río de Janeiro, 1936.
30. Orozco, Elena. *La gramática desencuadernada*. México, Eds. El Caballito, 1985.
31. Seco, R. *Manual de gramática española*. Madrid, 1968.
32. Real Academia Española. *Esbozo de una Nueva Gramática de la Lengua Española.* Madrid, 1973.
33. Real Academia Española. *Ortografía de la Lengua Española*. Madrid, 1999.

REFERENCES

34. Apresian, Yu. D. *et al*. *Linguistic support of the system ETAP-2* (in Russian). Nauka, Moscow, Russia, 1989.
35. Beekman, G. "Una mirada a la tecnología del mañana". En *Computación e informática hoy*. Addison Wesley Iberoamericana, 1996.
36. Benveniste, E. *Problemas de lingüística general.* 7[a] ed., México, Editorial Siglo XXI, 1978.
37. Bolshakov, I. A. "Multifunction thesaurus for Russian word processing". *Proc. of 4th Conference on Applied Natural Language Processing*, Stuttgart, October 13-15,1994. pp. 200-202.
38. Bolshakov, I., S. N. Galicia-Haro. "Algoritmo para corregir ciertos errores ortográficos en español relacionados al acento". *Computación y Sistemas*, Vol. 1, No. 2, 1997, pp. 57-62.
39. *Compendio Xcaret de la escritura jeroglífica maya descifrada por Yuri V. Knórosov*. Universidad de Quintana Roo, el grupo Xcaret y el sello Vía Láctea, 1999.
40. *Inteligencia artificial. Conceptos, técnicos y aplicaciones*. Marcombo. Boixareu Editores. Barcelona—México, 1987.
41. Fellbaum, C. (ed.). *WordNet as Electronic Lexical Database*. MIT Press, 1998.
42. Gelbukh, A. *Using a Semantic Network Dictionary in Some Tasks of Disambiguation and Translation*. Technical report, Serie Roja, N 36. CIC, IPN, 1998, ISBN 970-18-1894-6, 23 pp.
43. Galicia-Haro, S. N., A. Gelbukh, I. A. Bolshakov. "Three Mechanisms of Parser Driving for Structure Disambiguation". Proc. CICLing-2001, Computational Linguistics and Intelligent Text Processing. *Lecture Notes in Computer Science*, N 2004, Springer-Verlag, 2001, pp. 190–192.
44. Gelbukh A., G. Sidorov, and A. Guzmán-Arenas. "A Method of Describing Document Contents through Topic Selection", *Proc. of SPIRE'99*, International Symposium on String Processing and

Information Retrieval, Cancun, Mexico, September 22-24, 1999, IEEE Computer Society Press, pp. 73–80.

45. Gelbukh A., G. Sidorov, and A. Guzmán-Arenas. “Use of a weighted topic hierarchy for text retrieval and classification”. In: Václav Matoušek *et al*. (eds.). *Text, Speech and Dialogue*. Proc. TSD-99. *Lecture Notes in Artificial Intelligence*, No. 1692, Springer-Verlag, 1999, pp. 130–135.
46. Guzmán-Arenas A. “Finding the main themes in a Spanish document”. *Journal Expert Systems with Applications*, Vol. 14, No. 1/2. Jan/Feb 1998, pp. 139-148.
47. Hausser, Ronald. *Foundations of computational linguistics: man-machine communication in natural language*. Springer Verlag, 1999.
48. Lara, L.F., *et al*. *Investigaciones lingüísticas en lexicografía.* El Colegio de México, México, 1989.
49. McKeown, Kathleen. *Text generation.* Cambridge University Press, Cambridge, 1985.
50. Mel’čuk, I. A. *Experience in theories of Meaning ⇔ Text linguistic models* (in Russian). Nauka, Moscow, Russia, 1974.
51. Pustejovsky, J. *The generative lexicon.* MIT Press. Cambridge, Massachusetts - London, 1995.
52. *Rabbit in the Moon* (information about Mayan culture): www.halfmoon.org.
53. Translation group of the Information Science Institute of the University of South California, www.isi.edu/natural-language/GAZELLE.html.
54. Tveter, D. R. *The Pattern Recognition Basis of Artificial Intelligence*. IEEE Computer Society. Los Alamitos, CA, 1998.
55. Vossen, P. (ed.). *EuroWordNet General Document*. Version 3: www.hum.uva.ln/~ewn.

APPENDICES

SOME SPANISH-ORIENTED GROUPS AND RESOURCES

HERE WE PRESENT a very short list of groups working on Spanish, with their respective URLs, especially the groups in Latin America. The members of the RITOS network (emilia.dc.fi.udc.es /Ritos2) are marked correspondingly. In addition, we give a few URLs of dictionaries or directories of Spanish resources, as well as sites of general linguistic interest.

Scientific groups and associations

- AMPLN, Mexican Association for Natural Language Processing: www.ampln.org.
- SEPLN, Spanish Society for Natural Language Processing: www.sepln.org.
- Center for Computing Research (CIC, *Centro de Investigación en Computación*), National Polytechnic Institute (IPN, *Instituto Politécnico Nacional*), Laboratory of Natural Language, Mexico City, Mexico [RITOS]: www.cic.ipn.mx, www.Gelbukh.com; gelbukh @ gelbukh . com (gelbukh@cic.ipn.mx).
- National Institute for Astrophysics, Optics and Electronics (INAOE, Instituto Nacional de Astrofísica, Óptica y Electrónica), Aurelio López López and Manuel Montes y Gómez's group, Puebla, Mexico: www. inaoep. mx, cseg. inaoep. mx /~allopez, ccc.inaoep.mx/ ~mmontesg; allopez@GISC1.inaoep.mx, mmontesg@inaoep.mx.
- Institute for Investigation in Applied Mathematics and Systems (IIMAS, Instituto de Investigación en Matemáticas Aplicadas y en Sistemas), National Autonomous University of Mexico (UNAM) [RITOS], Luis Pineda Cortés's group, Mexico City, Mexico:

www.iimas.unam.mx, leibniz.iimas.unam.mx/~luis, luis@leibniz.iimas.unam.mx.

- Benemérita Universidad Autónoima de Puebla, the group of Héctor Himénez Salazar.
- National Autonomous University of Mexico (UNAM) [RITOS], Linguistic Engineering Group led by Gerardo Sierra Martínez: iling.torreingenieria.unam.mx.
- Center for Linguistic and Literature Studies (*Centro de Estudios Lingüísticos y Literarios*), El Colegio de México, Luis Fernando Lara's group, Mexico City, Mexico: www.colmex.mx/centros/cell/default.htm, lara@colmex.mx.
- Autonomous University of Tlaxcala, Mexico; the group led by Heriberto Cuayáhuitl Portilla.
- Computer Science Department, University of Chile, Ricardo Baeza Yates's group, Santiago de Chile, Chile: www.dcc.uchile.cl, www.dcc.uchile.cl/ ~rbaeza, rbaeza@dcc.uchile.cl.
- Department of Languages and Information Systems (LSI, *Departament de Llenguatges i Sistemes Informàtics*), Polytechnic University of Catalonia (UPC, *Universitat Politécnica de Catalunya*), Horacio Rodriguez Hontoria's group: www.lsi.upc.es, horacio@lsi.upc.es.
- *Facultad de Ingeniería*, *Instituto de Computación*, Montevideo, Uruguay [RITOS]: www.fing.edu.uy.
- Group of Data Structures and Computational Linguistics (*Grupo de Estructuras de Datos y Lingüística Computacional*), Spain: protos.dis.ulpgc.es.
- Research Group on Automatic Text Generation and Discourse Processing, Spain: www.ucm.es/info/atg.
- Reference Center of Linguistic Engineering (*Centre de Referència en Enginyeria Lingüística*): www.cesca.es/crel.
- Spanish Society of Natural Language Processing (*Sociedad Española para el Procesamiento del Lenguaje Natural*): gplsi.dlsi.ua.es/SEPLN.

- *Universidad Autónoma de Madrid* (UAM), *Facultad de Filosofía y Letras*, Laboratorio de Lingüística Informática.
- *Universidad Nacional de Educación a Distancia* (UNED), Madrid, Spain [RITOS]: www.uned.es.
- *Universidad Mayor de San Simón*, Cochabamba, Bolivia [RITOS]: www.umss.edu.bo.
- *Universidade da Coruña*, *A Coruña*, Spain [RITOS]: www.udc.es, brisaboa@udc.es.
- *Universitat de Barcelona* (UB), Toni Martí's group.
- *Pontifícia Universidade Católica do Rio Grande do Sul*, Porto Alegre, Brazil [RITOS]: www.pucrs.br, vera@kriti.inf.pucrs.br.

Resources and directories of resources

- Association for Computational Linguistics: www.aclweb.org.
- Archive of linguistic papers: www.cs.columbia.edu/~radev/u/db/acl.
- Compilers and languages (COLE): coleweb.dc.fi.udc.es.
- Dictionaries and encyclopedias: www.ucm.es/BUCM/cps/eng/0411.htm.
- Homepage of Spanish language (*Página de la Lengua Española*): www.latintop.com/espannol.
- Homepage of Spanish language (*Página del Idioma Español*): www.el-castellano.com.
- The Linguist List: www.linguistlist.org.
- Virtual Center Cervantes (Centro Virtual Cervantes): cvc.cervantes.es/portada.htm.

Some International Conferences

The conferences with the Proceedings published as issues as the journal Lecture Notes in Computer Science, Springer-Verlag, are marked in boldface.

- ACL (Association for Computational Linguistics), www.aclweb. org.
- **CICLing** (Computational Linguistics and Intelligent Text Processing), www.CICLing.org.
- COLING (Computational Linguistics), see www.dcs.shef.ac.uk/ research/ilash/iccl.
- **DEXA** (Databases and Expert Systems Applications), www.dexa. org.
- **NLDB** (Applications of Natural Language to Information Systems), www.nldb.org.
- NLPKE (Natural Language and Knowledge Engineering).
- RANLP (Recent Advances in Natural Language Processing), see, for example, lml.bas.bg/ranlp2003.
- SEPLN (Spanish Society for Natural Language Processing), www.sepln.org.
- **TSD** (Text, Speech and Dialogue), see, for example, nlp.fi.muni.cz/ tsd2004.

Additional materials and resources can be found on the webpage of this book, www.Gelbukh.com/clbook. Other publications by the authors of this book can be found at www.Gelbukh.com.

ENGLISH-SPANISH DICTIONARY OF TERMINOLOGY

actant actuante
ambiguity ambigüedad
analysis análisis
analyzer analizador
CFG see *context-free grammars*
circonstant circunstante
coherence coherencia
constituency constituyencia
constituency tree árbol de constituyencia
constituent constituyente
consonant consonante
context-free grammars gramáticas libres de contexto
deep structure estructura profundo
dependency dependencia
dependency tree árbol de dependencia
dialectology dialectología
discourse discurso
Generalized Phrase Structure Grammar Gramática Generalizada de Estructura de Frase
generation generación
generative grammars gramáticas generativas
government pattern patrón de rección (régimen)
GPSG see *Generalized Phrase Structure Grammar*
grammar checking revisión de gramática
grammatic case caso gramatical
Head-driven Phrase Structure Grammar Gramática de Estructura de Frase Manejada por Núcleo
historic linguistics lingüística histórica
homonym homónimo
homonymy homonimía
HPSG see *Head-driven Phrase Structure Grammar*
hyphenation división de palabras con guiones
information retrieval recuperación de información
IRS, information retrieval system sistema de recuperación de información
interpretation interpretación
level of representation nivel de representación
lexeme lexema
lexicography lexicografía
linear structure of text estructura lineal de texto
linguistic sign signo lingüístico
logical predicate predicado lógico

main topic of a document tema principal del documento
mathematical linguistics lingüística matemática
mathematical logic lógica matemática
meaning significado
Meaning ⇔ Text model modelo Significado ⇔ Texto
Meaning ⇔ Text Theory Teoría Significado ⇔ Texto
morph morfo
morphology morfología
morphosyntactic morfosintáctico
MTT see *Meaning ⇔ Text Theory*
natural language interface interfaz del lenguaje natural
natural language understanding comprensión de lenguaje natural
nested structure estructura anidada de texto
noun sustantivo
noun phrase sintagma nominal
onomatopoeia onomatopeya
optical character recognition reconocimiento óptico de caracteres
oriented labeled graph grafo orientado (dirigido) con etiquetas
orthographic error error ortográfico
parser analizador sintáctico (párser)
part of speech categoría gramatical
phonetic alphabet alfabeto fonético
phonetic symbol símbolo fonético
phonetic transcription transcripción fonética
phonology fonología
phrase frase, sintagma
phrase structure estructura de frase
polysemic polisemántico
polysemy polisemia
pragmatics pragmática
psycholinguistics psicolingüística
reference referencia
search engine motor de búsqueda
semantic network red semántica
semantic representation representación semántica
semantics semántica
seme sema
semiotics semiótica
sides of linguistic sign lados del signo lingüístico
sign signo
signified significado
signifier significante

sociolinguistcs lingüística sociológica
speech recognition reconocimiento del habla
spelling error error de mecanografía, de ortografía
spell checking revisión ortográfica
style checking revisión estilística
sublanguage sublenguaje
synonym sinónimo
synonymy sinonimia
syntactic analyzer analizador sintáctico
syntactic predicate predicado sintáctico
syntactic structure estructura sintáctica
syntactics sintáctica
syntax sintaxis
synthesis síntesis
synthesizer sintetizador
term término
text texto
text preparation preparación del texto
transformational grammars gramáticas transformacionales
translation traducción
unification unificación
typographic error see *spelling error*
valency valencia
vowel vocal
wordform forma (gramatical) de la palabra

INDEX OF ILLUSTRATIONS

INDEX OF AUTHORS, SYSTEMS, AND TERMINOLOGY

CAN COMPUTERS meaningfully process human language? If this is difficult, why? If this is possible, how? This book introduces the reader to the fascinating science of computational linguistics and automatic natural language processing, which combines linguistics and artificial intelligence.

The main part of the book is devoted to the explanation of the inner working of a linguistic processor, a software module in charge of translating natural language input into a representation directly usable traditional artificial intelligence applications and, vice versa, of translating their answer into human language.

Overall emphasis in the book is made on a well-elaborated, though—for a number of historical reasons—so far little-known in the literature computational linguistic model called Meaning ⇔ Text Theory. For comparison, other models and formalisms are considered in detail.

The book is mainly oriented to researchers and students interested in applications of natural language processing techniques to Spanish language. In particular, most of the examples given in the book deal with Spanish language material—which is a feature of the book distinguishing it from other books on natural language processing. However, our main exposition is sufficiently general to be applicable to a wide range of languages.

Specifically, it was taken into account that many readers of the book will be Spanish native speakers. For them, some comments on the English terminology, as well as a short English-Spanish dictionary of technical terms used in the book, were included. Still, reading the book in English will help Spanish-speaking readers to become familiar with the style and terminology used in the scientific literature on the subject.



IGOR A. BOLSHAKOV

was born in Moscow, Russia, in 1934. He obtained his M.Sc. degree in physics in 1956 from the Department of physics of the Moscow State "Lomonossov" University, Ph.D. degree in information technology in 1961 from the VYMPEL Institute, Moscow, Russia, and D.Sc. degree in computer science in 1966 from the same institute. He received the National Award of USSR in Science and Technology in 1989. Since 1996, he works for the Natural Language and Text Processing Laboratory of the Computing Research Center, National Polytechnic Institute, Mexico City. He is National Researcher of Mexico of excellence level III, author of more than 200 publications on theory of radars, theory of probability, and computational linguistics.

Email: igor@cic.ipn.mx



ALEXANDER F. GELBUKH

was born in Moscow, Russia, in 1962. He obtained his M.Sc. degree in mathematics in 1990 from the Department of mechanics and mathematics of the Moscow State "Lomonossov" University and Ph.D. degree in computer science in 1995 from the All-Russian Institute for Scientific and Technical Information. Since 1997 he is the head of the Natural Language and Text Processing Laboratory of the Computing Research Center, National Polytechnic Institute, Mexico City. He is academician of the Mexican Academy of Sciences, National Researcher of Mexico of excellence level I, distinguished lecturer of the ACM, founder of the Mexican Association for Natural Language Processing and the CICLing international conference series, author of more than 250 publications on computational linguistics. Currently he is Distinguished Visiting Professor at Chung-Ang University, Seoul, Korea.

Webpage: www.Gelbukh.com

CAN COMPUTERS meaningfully process human language? If this is difficult, why? If this is possible, how? This book introduces the reader to the fascinating science of computational linguistics and automatic natural language processing, which combines linguistics and artificial intelligence.

The main part of the book is devoted to the explanation of the inner working of a linguistic processor, a software module in charge of translating natural language input into a representation directly usable traditional artificial intelligence applications and, vice versa, of translating their answer into human language.

Overall emphasis in the book is made on a well-elaborated, though—for a number of historical reasons—so far little-known in the literature computational linguistic model called Meaning ⇔ Text Theory. For comparison, other models and formalisms are considered in detail.

The book is mainly oriented to researchers and students interested in applications of natural language processing techniques to Spanish language. In particular, most of the examples given in the book deal with Spanish language material—which is a feature of the book distinguishing it from other books on natural language processing. However, our main exposition is sufficiently general to be applicable to a wide range of languages.

Specifically, it was taken into account that many readers of the book will be Spanish native speakers. For them, some comments on the English terminology, as well as a short English-Spanish dictionary of technical terms used in the book, were included. Still, reading the book in English will help Spanish-speaking readers to become familiar with the style and terminology used in the scientific literature on the subject.

Diseño de portada: Guadalupe Villa Ramírez






INSTITUTO POLITÉCNICO NACIONAL
UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
FONDO DE CULTURA ECONÓMICA


ISBN 970-36-0147-2
9 789703 601472